LAUREA

Shibboleth – tunnistautumisjärjestelmä korkeakouluille

Helenius, Velimatti & Mohrdar-Ghaemmaghami, Arsalan

2009 Leppävaara

Laurea-ammattikorkeakoulu
Laurea Leppävaara

Shibboleth – tunnistautumisjärjestelmä korkeakouluille

Velimatti Helenius
Arsalan Mohrdar-Ghaemmaghami
Tietojenkäsittelyn koulutusohjelma
Opinnäytetyö
Joulukuu, 2009

Laurea-ammattikorkeakoulu **Tiivistelmä**
Laurea Leppävaara
Tietojenkäsittelyn koulutusohjelma

Velimatti Helenius, Arsalan Mohrdar-Ghaemmaghami

Shibboleth – tunnistautumisjärjestelmä korkeakouluille

Vuosi 2009 Sivumäärä 95

Nykypäivän korkeakouluissa on tullut esiin mahdollisuus hyödyntää toisten korkeakoulujen tietoresursseja. Tietoresursseihin pääsee helpoimmin käsiksi internetin välityksellä, mutta tämä skenaario vaatii tietoturvallisen ohjelmistoratkaisun toimiakseen hyväksyttävästi. On myös yksinkertaisuuden kannalta tärkeää, että opiskelija pääsee toisten koulujen materiaaleihin käsiksi omalla käyttäjätunnuksella ja salasanallaan.

Tämä opinnäytetyö käsittelee ohjelmistoa nimeltä Shibboleth. Shibboleth antaa opiskelijoille mahdollisuuden päästä toisten korkeakoulujen materiaaleihin käsiksi oman koulunsa käyttäjätunnuksella ja salasanalla.

Opinnäytetyön toimeksiantajana oli Laurea-ammattikorkeakoulu. Toimeksiantona oli oppia ja dokumentoida Shibboleth-järjestelmän asennus ja tämän konfigurointi sekä pohtia sen sovellutusta korkeakouluissa.

Shibboleth-järjestelmässä on ylläpitäjän kannalta puutteita, joten tavoitteena oli myös kartoittaa niitä sekä tehdä niiden pohjalta kehitysehdotuksia.

Projektin tuloksena luotiin täysin toimiva Shibboleth-ympäristö Laurean Leppävaaran IT-infon tiloihin. Asennus ja konfigurointi dokumentoitiin, jotta järjestelmän käyttöönotossa saatavilla on yksiselitteiset ja tarkat ohjeet.

Asiasanat : Shibboleth, Tunnistautuminen, Korkeakoulu, Federaatio

Laurea University of Applied Sciences **Abstract**
Laurea Leppävaara
Business Information Technology Programme

Velimatti Helenius, Arsalan Mohrdar-Ghaemmaghami

Shibboleth – an identification system for higher education institutes

Year 2009 Pages 95

Today's higher education institutes have the opportunity to take advantage of other higher education institutes' data resources. They are most easily accessed via internet, but this scenario requires a secure software solution in order to work acceptably. It is also important for the sake of simplicity that a student can access other schools' materials with his own username and password.

This thesis discusses software by the name of Shibboleth. Shibboleth gives the students an opportunity to access other higher education institutes' materials with the students' own username and password.

The client for this thesis was Laurea University of Applied Sciences. The client wanted the writers to learn and document how to install and configure the Shibboleth system as well as to reflect on the application of the system in higher education institutes.

In the Shibboleth system there are shortcomings from the perspective of the administrator, so the goal was also to try to map them and make development propositions based on them.

As the result of the project, a fully working Shibboleth environment was created in the premises of Laurea Leppävaara IT-info. The Installation and the configuration were documented so that when the system is deployed In Laurea, there are unambiguous and accurate instructions available.

Key words: Shibboleth, Identification, Higher education institute, Federation

Sisällys

1 Johdanto

Tulevaisuus tuo tullessaan uutta tekniikkaa ja teknologiaa. Tänä päivänä korkeakouluilla ja yliopistoilla on jokaisella omat oppimisalustansa ja työympäristöt, joihin kirjaudutaan erillisillä niille varatuilla tunnuksilla. Salasanojen muistaminen ja moneen paikkaan erikseen kirjautuminen on vaivalloista ja hankalaa, varsinkin käyttäjän kannalta. Tunnukset vaihtelevat ja salasanat muuttuvat, mikä lisää vielä tunnusten muistamisen tuskaa. Shibboleth-järjestelmän tarkoitus on helpottaa käyttäjätunnistusta ja mahdollistaa kirjautuminen yhdellä tunnuksella eri työympäristöihin tai oppimisalustaan sekä mahdollistaa kertakirjautuminen. Shibboleth on erityisesti suunniteltu kirjautumiseen organisaatioiden ylitse. Tämän periaatteen mukaan on mahdollista kirjautua toisen organisaation ympäristöön oman organisaationsa tunnuksella. Tämänlaisen järjestelmän tarve on erittäin suuri tällä hetkellä. Kyseinen järjestelmä onkin jo käytössä monissa korkeakouluissa ja yliopistoissa.

Valitsimme opinnäytetyöaiheeksemme Shibboleth-käyttäjätunnistusjärjestelmän. Saimme toimeksiannon alun perin Laurea-Ammattikorkeakoulu Oy:n IT-palvelulta. Laureassa opiskelee vuosittain noin 8000 opiskelijaa joihin kuuluu sekä ylemmän että alemman amk-tutkinnon opiskelijoita.

Laurea-Ammattikorkeakoulun on tarkoitus siirtyä Shibboleth-käyttäjätunnistusjärjestelmään kuluvana vuotena. Projektimme tehtävänä olikin pilotoida Shibboleth-järjestelmän asennus, konfigurointi, ja dokumentointi. Tekemäämme pilotointia hyödynnetään tulevaisuudessa, kun Laurea-Ammattikorkeakoulu ottaa kyseisen järjestelmän käyttöönsä. Shibboleth-tunnistautumisjärjestelmä on käytössä monessa muussa organisaatiossa, joten vastaavia hankkeita on toteutettu kymmenissä eri organisaatioissa. Eri lähteistä saamamme muiden organisaatioiden asennuspäiväkirjat tai asennusohjeet eivät tyydyttäneet projektin jäseniä jonka takia pyrimme työstämään käyttökelpoiset asennusohjeet Shibboleth-järjestelmään ja sen käyttöönottoon. Pystyimme jonkin verran hyödyntämään muiden organisaatioiden asennuksia ja pilotointeja mutta kuitenkin itse asetukset piti räätälöidä oman ympäristön ja vaatimuksien mukaisiksi.

Kaikki projektin asennukset dokumentoitiin sekä projektiin liittyvät teoriapohjat selvitettiin kirjallisesti. Projektin valmiista lopputuloksista tehtiin selvitys sekä omat päätelmät. Lopuksi mietimme kehitysehdotuksia projektiin liittyen.

2 Käsitteet

2.1 IDP – Identity Provider

IdP eli Identity Provider on toinen Shibbolethin pääkomponenteista. IdP tarjoaa tunnistautumiseen vaadittavan käyttäjän identiteetin ja välittää sen eteenpäin.

2.2 SP – Service Provider

SP eli Service Provider on toinen Shibbolethin pääkomponentti, jonka tarkoitus on tarjota ja suojata sen ylläpitämää palvelua.

2.3 Shibboleth Installfest

Shibboleth Installfest on Shibbolethin eräänlainen testiympäristö. Sen avulla pyritään kuvaamaan shibbolethin asennus, konfigurointi ja toiminta mahdollisimman helposti ja pienellä työmäärällä. Installfest ympäristö pitää sisällään valmiiksi asennetun Shibboleth-järjestelmän, etukäteen konfiguroidut tiedostot ja valmiin tietokannan. Installfestin tarkoitus on opastaa Shibboleth-järjestelmän käyttöönotto.

2.4 XML

XML eli Extensible Markup Language on niin sanottu metakieli. Sen kehittäjä on World Wide Web Consortium. XML on hyvin lähellä HTML kieltä mutta sitä ei ole tarkoitettu sivunkuvauskieleksi vaan se on suunniteltu internet-käyttöön. Xml data koostuu tekstimuotoisesta informaatiosta jossa kuvataan esimerkiksi tiedon nimi, sijainti ja merkitys. (Ray 2003)

2.5 SAML

SAML eli Security Assertion Markup Language on standardi. Sitä käytetään autentikointiin ja tunnistautumiseen palvelimien välillä. Pääsääntöisesti SAML:ia käytetään kertakirjautumiseen joka liikkuu nimellä Single Sign-on. Käytännössä on mahdollista yhdellä kirjautumisella päästä useampaan internetissä olevaan palveluun sen sijaan että tarvitsisi kirjautua jokaiseen paikkaan erikseen. SAML standardi määrittelee tarvittavat tiedot joita tarvitaan turvallisuustietojen käsittelyyn. SAML tarjoaa tietoturvallisen viestienvaihdon eri osapuolten välillä. Autentikointia kontrolloidaan SAML-kannanottojen (engl.statement)avulla. nämä ovat:

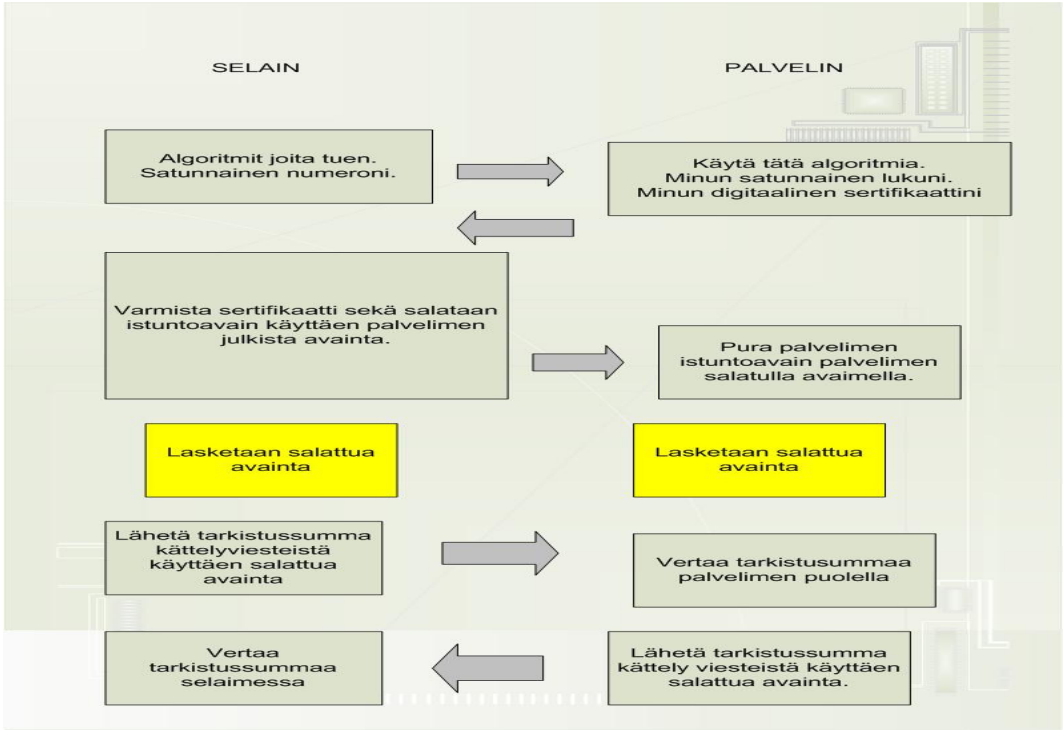
1. Authentication statements
2. Attribute statements

3. Authorization decision statements

Autentikaatio-kannanotoissa (authentication statements) välitetään SAML-lausunto halutulle palveluntarjoajalle. Attribuuttien kannanotto (attribute statements) hoitaa haluttujen attribuuttien välityksen palveluntarjoajalle pääsynvalvontaa varten. Viimeinen kannanotto koskee auktorisointitietoja. Se välittää tietoa käyttöoikeuksista halutussa alustassa. Kannanottoja käytetään SAML-kyselyillä valtuuttajan ja palveluntarjoajan välillä.
SAML rakentuu kolmesta eri kerroksesta jotka ovat väittämät (engl. assertions), protokollat, profiilit ja bindit (engl. bindings). (Oasis 2004)

2.6 SSL

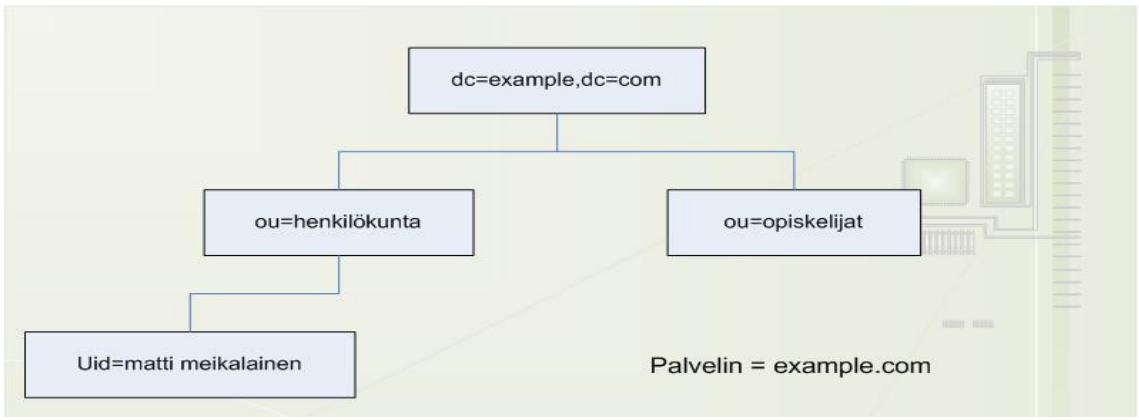
Yksi keino toteuttaa suojattu yhteys julkisen IP–verkon yli on SSL eli Secure Sockets Layer. Nykyään SSL on yksi suosituimmista salausmenetelmistä. SSL on alun perin kehitetty Netscapen internetselaimeen salausprotokollaksi, esim. WWW-sivujen suojaamiseen. SSL:n ominaisuuksiin kuuluu se että se ei rajoitu pelkästään HTTP protokollaan vaan toimii myös muissa TCP-protokollissa. Tällä hetkellä sitä käytetään mm. SMTP- , POP- , IMAP- ja LDAP-yhteyksien salaamiseen. SSL tarjoaa tehokkaan salauksen verkkoliikenteelle. Myös muut selaimet kuin Netscape tukevat SSL salausta. Protokollan tärkein tehtävä on viestien salaus, joten se voi salata käyttäjän ja palvelimen välistä liikennettä siten että myös käyttäjätunnukset ja salasanat pysyisivät salassa. Yksi tärkeistä ominaisuuksista joita SSL pitää sisällään on autentikointi sertifikaattien avulla. Näin voidaan olla varmoja että vastaanottaja tai lähettäjä on luotettu taho. Salaus tapahtuu käyttämällä jaettua avainta ja ennalta sovittua algoritmiä. SSL toimii portissa 443. (Hunt 2002)



Kuvio 1: SSL-tekniikan viestinvaihdot. (Cibernarium 2009)

2.7 LDAP

LDAP eli Lightweight Directory Access Protocol on eräänlainen protokolla hakemistopalvelulle. Se kehitettiin Michiganin yliopistossa. LDAP on tietokanta tai hakemisto jonne on tallenettu esim. nimi, osoitetiedot sekä mahdolliset käyttäjäoikeudet. LDAP käyttää TCP/IP-protokollaa verkon kautta. LDAP:n päätehtävä on käyttäjätunnistus. Käyttäjää tunnistettaessa palvelin tarkistaa vaadittavat attribuutit LDAP:n hakemistorakenteesta. LDAP siis voi pitää sisällään paljon muutakin kuin vain käyttäjätunnuksen ja salasanan. Pääsääntöisesti LDAP:ia käytetään varmennehakemistona, henkilötietojen säilömiseen sekä verkon yli tunnistautumiseen.



Kuvio 2: Esimerkki LDAP puun rakenteesta

LDAP:n rakenne on puumainen ja sen juuri on nimetty DNS:n mukaan. Nimi erotetaan pilkuilla eri verkkoaluekomponentteihin. Tässä tapauksessa example.com muuttuu dc=example, dc=com. Kun LDAP-puun juuri on nimetty, voidaan sinne sijoittaa eri organisaatioyksiköt kuten henkilökunta ja opiskelijat. Organisaatioyksiköiden alle voidaan lisätä käyttäjiä vaikkapa uid:n eli käyttäjätunnuksen mukaan. LDAP hakemisto on hyvä suojata SSL:llä sekä suojata se anonyymeiltä hauilta. Tämä tarkoittaa sitä että kaikille jotka käyttävät LDAP-hakemistoa on luotava erilliset LDAP-tunnukset. Yleensä ohjelmiin jotka tukevat LDAP-protokollaa, on mahdollista määrittää erikseen Bind-DN eli tunnus jolla ohjelma käyttää ja selaa hakemistoa. Tällaisia ohjelmia ovat mm. Shibboleth, PubCookie sekä Apachen lisämooduli auth_ldap. (Howes, Smith, Good 2003)

2.8 OpenLDAP

OpenLDAP on ilmainen avoimen lähdekoodin toteutus lightweight directory access protokollaan. Se siis mahdollistaa kyseisen protokollan käytön. Sen on toteuttanut OpenLDAP project-yhteisö. OpenLDAP:ssa on kolme pääkomponenttia. Näistä ensimmäinen on SLDAP joka on ns. stand-alone ohjelma. SLDAP pitää sisällään tarvittavat liitännäiset sekä työkalut. Toinen komponentti on kirjastot jotka mahdollistavat ldapin käyttöönoton. Viimeinen komponentti on itse client-työkalu joita ovat mm. Ldapsearch, ldapadd, ldapdelete. (OpenLDAP Foundation 2008)

2.9 JXplorer

JXplorer on ilmainen avoimen lähdekoodin LDAP-selain. Sen on alunperin on kehittänyt Computer Associates' eTrust Directory development lab. JXplorer on standardeja noudattava peruskäytön LDAP-selain joka voi selata ja muokata LDAP-hakemistoja, tai mitä tahansa x509 standardin mukaisia hakemistoja joissa on LDAP rajapinta. (Christopher Betts 2009)

2.10 CSC

CSC on opetusministeriön hallinnoima tieteen tietotekniikan keskus joka tarjoaa korkeakouluille, tutkimuslaitoksille ja yrityksille tietoteknistä tukea ja resursseja kuten esim. Shibboleth. (CSC Tietotekniikan keskus 2009)

2.11 TOMCAT

Tomcat on avoimen lähdekoodin toteutus Java-pohjaiselle sovelluspalvelimelle. Tomcat-palvelin ei ole rajoitettu vain tarjoamaan staattista html-sivua vaan se voi myös tarjota oh-

jelmia käyttäjien toiveiden mukaisesti. Tomcat:ia voidaan käyttää ns. stand-alonena tai Apahcen kanssa. (The Apache Software Foundation 2009)

2.12 APACHE

Apache on HTTP-palvelin, joka on avoimen lähdekoodin ohjelma. Apache HTTP-palvelin on suosituin ja tunnetuin Apache Software Foundationin ohjelma. Apache HTTP-palvelin on saatavilla useimmille käyttöjärjestelmille. Apache pystyy jakamaan vain staattisia tiedostoja HTTP-protokollan yli. Siihen on saatavilla useita erilaisia moduuleita, joilla voi tarvittaessa ladata ja muokata Apachea omien tarpeidensa mukaiseksi. Apache on internetin suosituin HTTP-palvelin. (The Apache Software Foundation 2009)

2.13 CentOS

CentOS on ilmainen avoimen lähdekoodin käyttöjärjestelmä joka perustuu Red Hat Enterprise Linuxiin. (The Community Enterprise Operating System 2009)

3 Käyttäjätunnistus

3.1 Nykytilanne

Suomessa korkeakoulut sekä yliopistot panostavat erilaisiin palveluihin joihin käyttäjät pääsevät käsiksi internetin kautta. Suurin osa internetissä tarjottavista palveluista vaatii käyttäjätunnistuksen. Käyttäjätunnistusta käytetään koska näin voidaan tuottaa henkilökohtaisempia palveluita henkilöille jotka järjestelmää käyttävät. Kun vanhoja palveluita lopetetaan ja uusia otetaan käyttöön, niin yleensä tunnuksetkin muuttuvat. Loppujen lopuksi tullaan siihen pisteeseen että käyttäjä tarvitsee monia eri tunnuksia moniin eri palveluihin. Tästä esimerkkinä se, että sähköpostinluku ja koulun kirjastosta lainaaminen tapahtuvat eri tunnuksilla. Tämä tuottaa suuria vaikeuksia käyttäjille ja järjestelmän ylläpitäjille. Ihanteellinen tilanne käyttäjän kannalta olisi, että olisi vain yhdet tunnukset joiden avulla pystytään kirjautumaan mihin tahansa palveluun organisaatiosta riippumatta. Käyttäjätunnuksien ja salasanojen karsimisen lisäksi organisaatioissa olisi mahdollista toteuttaa SSO(engl. Single Sign On)-kertakirjautumisjärjestelmä. SSO-periaatteen mukaan käyttäjän tarvitsee kirjautua vain kerran järjestelmään, jonka jälkeen hänellä on pääsy kaikkiin järjestelmiin ja palveluihin joita organisaatio pitää sisällään. Shibboleth käyttäjätunnistusjärjestelmä tarjoaa ratkaisun näihin edellä mainittuihin ongelmiin, lisää käyttäjäystävällisyyttä sekä tuo uusia mahdollisuuksia tunnistautumisen saralla.

WWW-sovellukset pystyvät tarjoamaan tarkempia ja henkilökohtaisempia palveluita jos käyttäjän muitakin tietoja kuin käyttäjätunnus on saatavilla, esim. henkilön status työpaikalla. Näiden tietojen perusteella voidaan tuottaa juuri tietylle henkilölle tarkoitettu kohdenäkymä

tietyssä palvelussa. Opiskelijoille eri kouluissa ovat varmasti tulleet tutuiksi opintosuoritteet ja niiden seuraaminen. Hyvänä esimerkkinä henkilökohtaisemmasta palvelusta on Winha-Wille opiskelijasivut joilla opiskelija pystyy seuraamaan suorituksiaan. Usein kuitenkin eri palveluihin kirjautumiseen riittää pelkkä taustamuuttujien tietäminen. Järjestelmään sisäänkirjautuessa palveluiden näkymät voitaisiin muokata erinäköiseksi riippuen siitä mitä koulutusohjelmaa opiskelija opiskelee. Joihinkin palveluihin taas tietyillä henkilöillä ei ole oikeutta kirjautua, esimerkkinä tästä voivat olla esim. talouspalvelun järjestelmät joihin muilla kuin talouspalveluiden työntekijöillä ei ole asiaa. Eri käyttäjien tarkempien tietojen huomioiminen vaatii käyttäjätunnistuksen. (Linden 2004)

Käyttäjätunnistus pitää sisällään kolme asiaa; henkilöllisyyden varmistaminen sekä henkilötietojen ja käyttöoikeuksien hallinta. Tässä kohtaa mukaan astuu käyttäjähallinto. Henkilöllisyyden varmistaminen eli autentikointi (engl.authentication)tehdään yleensä perinteisesti salasanalla, vaikkakin parempia keinoja turvallisuuden kannalta olisivat esim. kertakäyttösalasanat tai varmenteet. Nykypäivän teknologia kuitenkin tarjoaa siihen monia eri mahdollisuuksia, esim. sormenjälkitunnisteet ja biopassit. Henkilötietojen hallinnointi pitää sisällään tietojen ajantasalla pitoa ja tietojen oikeellisuuden varmistamista kuten esim. henkilön nimitietojen sekä sähköpostiosoitteiden. Käyttöoikeuksien hallinta eli auktorisointi (engl. authorization) on ikään kuin jatke henkilötietojen hallinnoinnille. (Linden 2004)

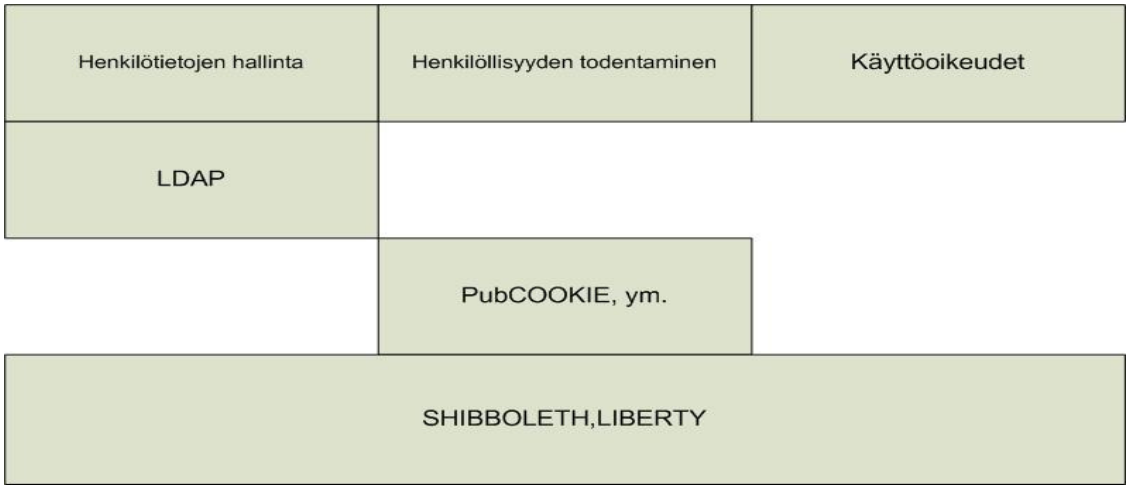
Käyttäjätietojen ylläpito on työlästä organisaatiossa joka pitää sisällään tuhansia henkilötietoja. Muutoksia tapahtuu jatkuvasti nimissä ja osoitteissa, ja jos näitä pitää korjata moneen eri järjestelmään jää se helposti tekemättä tai inhimillisesti unohtuu. Jos tietoja on ainoastaan yhdessä paikassa, niin sitä kannattaa hyödyntää myös tietojärjestelmien käyttäjähallinnossa. Yhden perusrekisterin hyväksikäyttäminen käyttäjätunnistuksessa laitoksen sisällä tuo uusia ratkaisuja käyttäjähallinnon tekniikoissa. Itse käyttäjälle tämä näkyy tietojen oikeellisuutena ja paikkaansa pitävyytenä mutta se tarjoaa myös yhden käyttäjätunnuksen ja näin ollen vain yhden kertakirjautumisen. (Novell 2007)

Yliopistojen yhdistyminen ja erilaisten federaatioiden synty mahdollistaa sen että palvelujen tuottajat ja käyttäjät eivät välttämättä edusta samaa organisaatiota. Esimerkiksi opiskelijat voivat suorittaa erilaisia verkkokursseja jotka suoritetaan toisen organisaation tai korkeakoulun oppimisalustassa. Tietyn koulutusohjelman opiskelijat voivat hyödyntää samaa oppimisalustaa riippumatta siitä mikä heidän kotiorganisaationsa on. Organisaatiorajat ylittävä tunnistautuminen asettaa omat haasteensa ja kriteerinsä käyttäjän tunnistautumiselle. (Linden 2007)

3.2 Toteutustapoja

Käyttäjätunnistukseen on kehitetty erilaisia kaupallisia sekä ei kaupallisia eli ns. avoimen lähdekoodin järjestelmiä. Eri järjestelmät suorittavat tunnistamisen eri tavoin.

Alla olevassa kuvassa on kuvattu eri WWW-tunnistautumisen tekniikoita sekä kuvattu niiden soveltuvuusaloja, joita on käytössä suomalaisissa korkeakouluissa (KUVIO 3).



Kuvio 3: Eri www-tunnistautumistekniikat ja niiden soveltuvuus. (Linden 2007)

LDAP on Windows AD:n kanssa perinteinen käyttäjähallintamenetelmä. LDAP:iin tukeutuvat käyttäjätunnistukset toimivat WWW-palvelimissa. Niin kuin edellä mainittiin, on Apache-palvelimelle mahdollisuus ladata erilaisia moduuleita jotka mahdollistavat LDAP-autentikoinnin (ldap_auth). (Linden 2007)

Ongelmat kuitenkin alkavat siinä vaiheessa, kun pitää esimerkiksi sallia pääsy LDAP:iin muille kuin oman organisaation jäsenille. Toisen organisaation www-palvelimen ottaessa yhteyttä käyttäjätietokantaan pitää olla varmuus siitä että yhteys on turvallinen. Miten esimerkiksi käyttäjän salasana kulkee viestinvaihdon aikana? Onko yhteys selaimen ja WWW-palvelimen välillä salattu? Huomioon pitää myös ottaa mitä tietoja käyttäjistä saa lähettää niitä pyytäville palvelimille.

PubCookie on hyvä esimerkki kertakirjautumisjärjestelmästä jolla ohitetaan edellisessä kappaleessa mainitut pulmat. PubCookie-ratkaisussa kaikki käyttäjätiedot sijoitetaan yhdelle luotetulle WWW-pavelimelle, jonka turvallisuutta ja ylläpitoa valvoo sen oma organisaatio. PubCookie-tekniikkaa voidaan käyttää turvallisin mielin eikä salasanan paljastumista tarvitse pelätä koska kaikki siihen tukeutuvat WWW-palvelimet ohjaavat aina kirjautumispyynnöt luotetulle autentikointipalvelimelle. PubCookie on alun perin kehitetty Washingtonin yliopistossa

ja kyseinen järjestelmä on avoimen lähdekoodin toteutus. Vastaavia kaupallisia toteutuksia on monia. Kuten monilla järjestelmillä niin myös PubCookiella on rajoituksensa, se hoitaa vain käyttäjän varmistamisen. PubCookieta käyttävä WWW-palvelin saa luotetulta autentikointipalvelimelta vain tiedon tunnistettavan käyttäjätunnuksesta. Käyttäjän muut tiedot sekä käyttöoikeudet on selvitettävä muulla keinolla, joka taas tuottaa uutta lisätyötä. (University of Washington 2007)

Shibboleth-tekniikka on erityisesti suunniteltu organisaatiorajojen ylittävään kirjautumiseen, joten siinä on otettu huomioon kaikki siihen vaadittavat kriteerit koskien turvallisuutta, todentamista ja oikeuksien jakamista. Tunnistautuminen joka tapahtuu yli organisaationrajojen vaatii että sille on asetettava erilaisia kriteerejä ja säännöksiä. Federaatioon liittyneiden osapuolien on sovittava organisaatioiden yhteisistä pelisäännöistä. Se miten jokaisen oman organisaation sisällä tunnistaudutaan ei ole merkityksellistä vaan se että organisaatiot käyttävät samaa tekniikkaa hyödyntäessään federaation sisäistä tunnistautumista. Korkeakoulut ja yliopistot Suomessa rakentavat tällä hetkellä yhteistä käyttäjätunnistuksen infrastruktuuria eli HAKA-infrastruktuuria. Kyseinen infra nojautuu Shibboleth-tekniikkaan. Tavoitteena on helpottaa organisaatioidenvälistä tunnistautumista ja tarjota palveluita muillekin kuin oman organisaation sisällä työskenteleville. Suurimman hyödyn tekniikasta toivotaan saavan virtuaaliyliopistojen, -ammattikorkeakoulujen sekä kirjastojen. Shibboleth-järjestelmää voitaisiin myös käyttää organisaation sisäiseen tunnistautumiseen.(CSC Tietotekniikan keskus 2009)

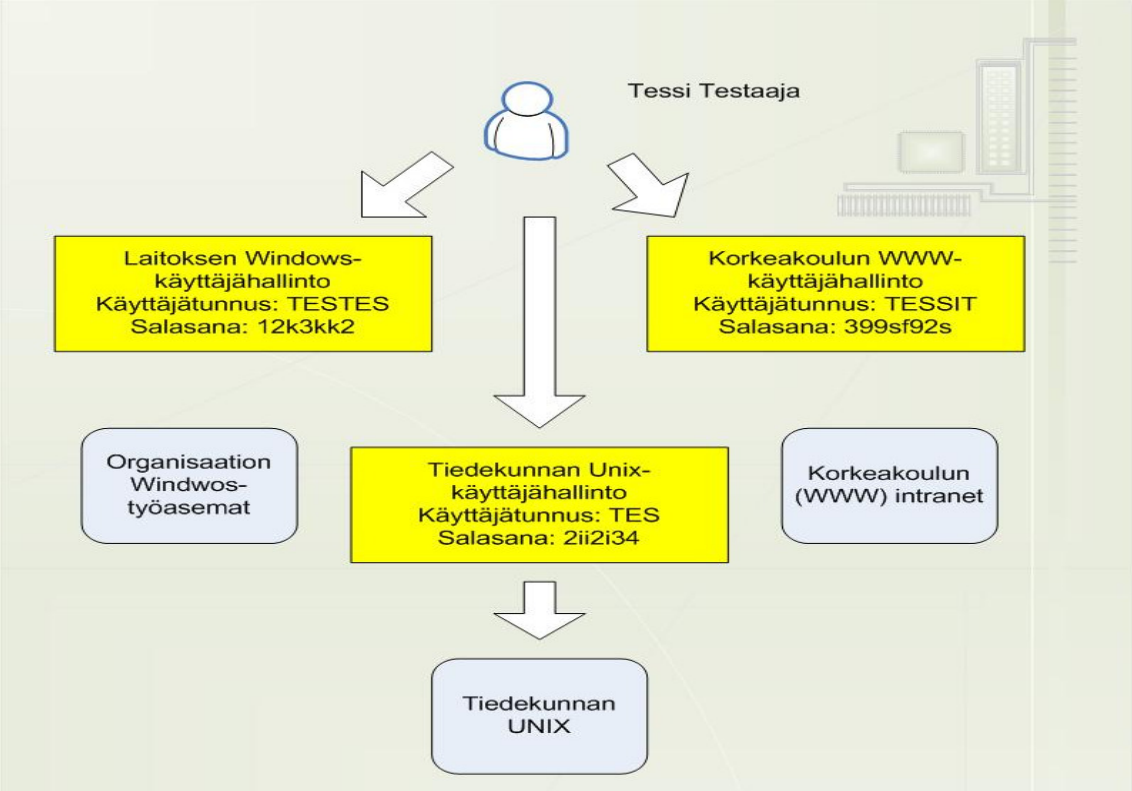
4 Käyttäjähallinto

Käyttäjähallintoa on hankala määrittää, koska se on järjestelmä jossa on osa tekniikkaa sekä osa ihmistä. Parempi kuvaus voisi olla että käyttäjähallinto on palvelu organisaation sisällä. Sillä tarkoitetaan niitä toimintaprosesseja ja tekniikoita joilla organisaatio pitää kirjaa järjestelmien käyttäjistä ja heidän käyttöoikeuksistaan. Käyttäjähallinnon tehtävä on valvoa eri palveluita ja niiden toimintaa, joten se koostuu erilaisista komponenteista. Käyttäjähallintoa on vaikea nähdä toiminnassaan, sillä se näkyy silloin vasta kun se ei toimi. Esimerkkinä; käyttäjä yrittää kirjautua palveluun mutta syystä tai toisesta kirjautuminen ei onnistu. Kun käyttäjähallinto toimii oikein, se tarjoaa joustavan ja toimivan pääsyn tietojärjestelmiin ja palveluihin. (Laaksonen, Nevasalo, Tomula. 2006)

4.1 Järjestelmäkohtainen käyttäjähallinto

Yleisesti katsottuna korkeakoulujen eri tietojärjestelmien ja palveluiden käyttäjähallinto on toteutettu järjestelmäkohtaisesti. Tämä siis tarkoittaa sitä että käyttäjällä monta eri tunnusta, yksi jokaiseen eri tietojärjestelmään tai palveluun jotka ovat toisistaan riippumattomia.

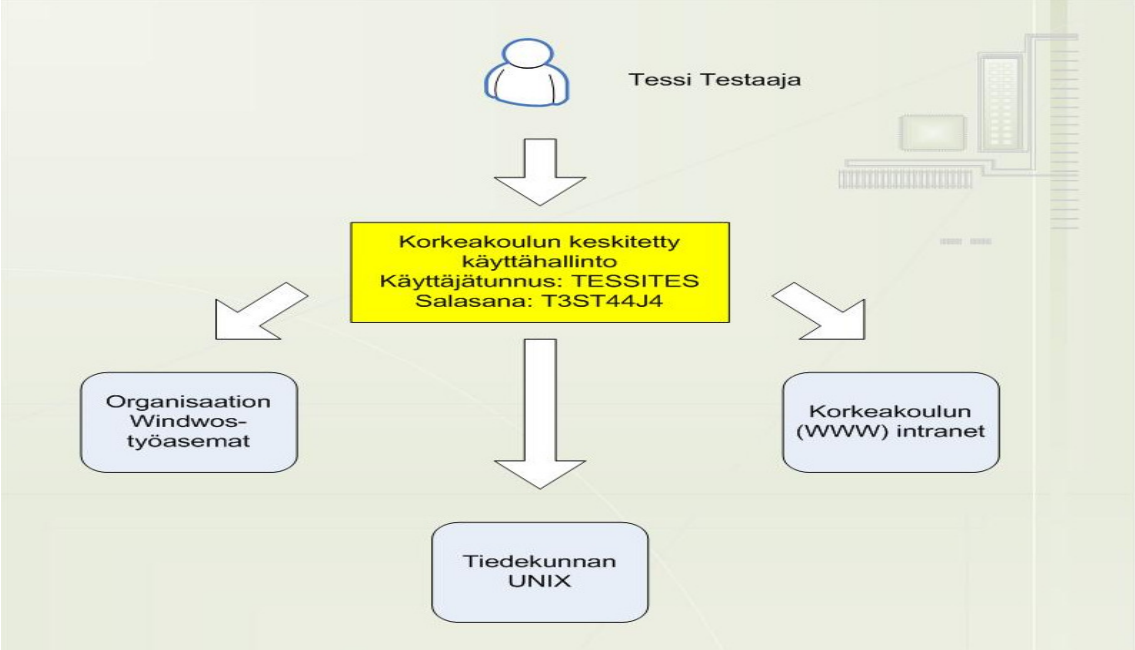
Alla on kuvattu esimerkki järjestelmäkohtaisesta käyttäjähallinnosta.(CSC Käyttäjähallinto korkeakoulussa) (KUVIO 4)



Kuvio 4:Järjestelmäkohtainen käyttäjähallinto. (CSC käyttäjähallinto korkeakoulussa)

4.2 Korkeakoulukohtainen Käyttäjähallinto

Kun käyttäjähallinto on korkeakoulussa keskitetty yhteen paikkaan, saadaan käyttöön vain yksi tunnus, jolla on mahdollisuus kirjautua eri palveluihin. Tämä tarkoittaa sitä että käyttäjien tietoja pidetään vain yhdessä paikassa. (CSC Käyttäjähallinto korkeakoulussa) (KUVIO 5)



Kuvio 5: Korkeakoulukohtainen käyttäjähallinto. (CSC käyttäjähallinto korkeakoulussa)

Keskitetyssä (korkeakoulukohtaisessa) käyttäjähallinnossa käyttäjällä on siis vain yksi tunnus mutta mahdollisesti käyttöoikeus useaan järjestelmään, riippuen tietenkin onko kyseiselle käyttäjälle annettu oikeuksia kirjautua eri järjestelmiin tunnuksellaan. Käyttäjähallinnon keskittämisen höytyjä ovat mm.

- Päällekkäinen ylläpitotyö vähenee eli tietoja ei tarvitse vaihdella eri järjestelmiin erikseen.
- Tiedot ovat ajan tasalla samassa paikassa, eli käyttäjien tietoja ei ole monessa eri paikassa jolloin virheellisten tietojen mahdollisuus vähenee.
- Käyttömukavuus, ei tarvitse olla monia eri tunnuksia.
- Turvallisuus, ei monia tunnuksia ja salasanoja ylöskirjoitettuna jonkin paperilehtiön kanteen.

Vaikka organisaation käytössä olisikin toimiva tekniikka niin se yksinään ei riitä. Käyttäjähallintoon liittyy paljon muutakin kuin pelkästään toimivat järjestelmät. Käyttäjähallinto vaatii toimiakseen yhteiset toimintamallit ja pelisäännöt organisaation sisällä. Organisaation käyttöoikeuksia pitää valvoa ja käyttäjän pitää olla tästä asiasta tietoinen. Esimerkiksi opiskelijan aloitettua oppilaitoksen kirjoilla, hänelle luodaan tunnukset järjestelmään josta ne sitten monistetaan eteenpäin eri järjestelmiin. Kun kyseinen henkilö eroaa organisaatiosta tai hän ei enää täytä valtuuksien kriteereitä, hänen tietonsa ja oikeutensa tulee poistaa käytöstä. Olisikin tärkeää että yliopiston tai ammattikorkeakoulun kaltaisessa organisaatiossa, jossa vaihtu-

vuus käyttäjien kesken on erittäin suuri ja käyttäjäkunta on hyvin samankaltainen, olisi käyttäjähallinnon perusprosessit käytössä keskeisimmissä järjestelmissä. Kun järjestelmät toimivat oikein ja hyvin perustasolla, järjestelmien laajentaminen suurempaan kokonaisuuteen helpottuu. (CSC Käyttäjähallinto korkeakoulussa)

4.3 Roolit

Tietojärjestelmissä on käyttäjille asetettu erilaisia rooleja, jotka määrittelevät käyttäjille erilaisia käyttöoikeuksia eri järjestelmiin ja palveluihin. Roolien hallinta helpottaa määrittämään ja hallinnoimaan käyttöoikeuksia eri järjestelmiin. Niiden avulla voidaan määrittää käyttäjäryhmiä eri järjestelmille ja palveluille. Roolien avulla on esimerkiksi määritetty että tietyn oppilaitoksen opiskelijat pääsevät kirjautumaan WinhaWille-opiskelijasivuille ja pystyvät tarkastelemaan sen hetkisiä suorituksiaan, kun taas opettajilla on niiden avulla sallittu oikeus kirjautua tenttienilmoittautusmispalveluun josta on mahdollista seurata tentteihin ilmoittautuneiden opiskelijoiden määrää. Ammattikorkeakouluissa ja yliopistoissa työskentelee myös paljon sellaisia käyttäjiä joilla on kaksoisrooli, eli he ovat opiskelijoita mutta kuuluvat samalla organisaation henkilökuntaan. Käyttäjähallinnon tehtävä on tarjota perusta roolitietojen käyttämiselle eri palveluissa. (CSC Käyttäjähallinto korkeakoulussa)

4.4 Virtuaaliverkostot

Erilaiset virtuaaliyliopistot ja oppimisalustat ovat lisääntyneet valtavalla vauhdilla. On mahdollista suorittaa kursseja avoimissa yliopistoissa ja muissa oppilaitoksissa virtuaalisesti. Tämä tarkoittaa sitä että käyttäjän tulisi olla mahdollista tunnistautua kotiorganisaationsa lisäksi myös opetusta tarjoaviin oppimisalustoihin ja virtuaaliympäristöihin. Ihanteellisena tapauksena tunnistautuminen tapahtuisi kotiorganisaatiossa josta hyväksytty tunnistautuminen siirtyisi kohdeoppimisalustaan. Korkeakoulurajojen ylittävää federaation sisäistä tunnistautumista rakennetaan korkeakoulujen HAKA-projektissa. (CSC Käyttäjähallinto korkeakoulussa)

4.4.1 Käyttäjähallinnon ja käyttäjätunnistusjärjestelmän vaatimuksia

Niinkuin edellä on mainittu, käyttäjähallintoon sekä käyttäjätunnistukseen on olemassa useita eri ratkaisuja jotka toimivat kaikki eri tavalla. Monet näistä järjestelmistä tarjoavat ratkaisun kauttaaltaan tai vain pienen osan koko hallinta/tunnistus prosessiin. Eri ratkaisuissa on omat vahvat ja heikot puolensa joten järjestelmää rakentaessa ja valitessa on hyvä kartoittaa oman organisaationsa tarpeet sekä minkälaisia vaatimuksia käyttäjähallinto organisaatiolle asettaa. Joitain perusvaatimuksia on hyvä asettaa tekniselle ratkaisulle. Seuraavassa muutamia esimerkkejä joita valmiissa toteutuksessa olisi hyvä olla:

- ”järjestelmän tulee tarpeen mukaan mahdollistaa kertakirjautuminen (Single Sign on) eri DNS-toimialueiden, SAML- tai muuta Web service security-standardia tukevien ulkopuolisten tahojen sekä sovellusten välillä.” (Laaksonen, Nevasalo, Tomula 2006 s.174-175)

- ”Järjestelmän tulee tukea eri autentikointimentelmiä, kuten LDAP, sertifikaatit, SAML (eri versiot), toimikortit, Kerberos, SecureID ja biometriset tunnistusmentelmät. Salasanoille ja muille autentikointimenetelmille tulee voida asettaa rajoituksia, jotka täyttävät organisaation vaatimukset.” (Laaksonen, Nevasalo, Tomula 2006 s.174-175)

- ”Järjestelmän tulee voida lukea autentikointi ja autorisointitietoja eri hakemistosta ja pystyä tarvittaessa hakemaan auktorisointitietoja myös ulkoisista järjestelmistä. Eri hakemistoissa olevat autentikointi ja auktosisointitiedot tulee voida normalisoida ja esittää yhtenä, ja niitä tulee voida hallita keskitetysti. Tarvittaessa oikeuksien hallinta tulee voida myös hajauttaa.” (Laaksonen, Nevasalo, Tomula 2006 s.174-175)

- ”Järjestelmän tulee olla SAML- ja WS-Federation-yhteensopivat.” (Laaksonen, Nevasalo, Tomula 2006 s.174-175)

- ”Järjestelmän tulee mahdollistaa erilaisten käyttörajoitusten toteuttaminen. Esim. joihinkin dokumentteihin ja järjestelmiin voi olla perusteltua sallia vain esim SecureID:n avulla tunnistettujen käyttäjien pääsy tai sallia joidenkin resurssien käyttö vain tiettyinä kellonaikoina ja tietyistä IP-osoitteista.” (Laaksonen, Nevasalo, Tomula 2006 s.174-175)

- ”Järjestelmän tulee olla virhesietoinen ja skaalautuva esim. kuormantasauksen avulla sekä mahdollistaa kaikkien järjestelmäkomponenttien välisen tiedonsiirron salaus.” (Laaksonen, Nevasalo, Tomula 2006 s.174-175)

- ”Järjestelmän tulee olla integroitavissa henkilöstöhallinnon järjestelmään tai muuhun taustajärjestelmään” (Laaksonen, Nevasalo, Tomula 2006 s.174-175)

- ”Käyttäjille tulee tarjota mahdollisuus omien tietojensa hallintaan itsepalveluperiaatteella. Käyttäjän tulee voida saattaa alulle jokin muutos mutta se pitää aina hyväksyä sopivan tahon toimesta” (Laaksonen, Nevasalo, Tomula 2006 s.174-175)

- ”Käyttäjien oikeudet tulee olla helposti raportoitavissa. Samoin tulee voida listata kaikki käyttäjät ja järjestelmät joilla on oikeus johonkin tiettyyn resurssiin. Näin ollen järjestelmä tukee laajaa ja tehokasta auditointia” (Laaksonen, Nevasalo, Tomula 2006 s.174-175)

- ”Oikeuksien muutosta ja käytöstä sekä muista tietoturvallisuuden kannalta oleellisista asioista tulee luotettavasti pitää keskitettyä lokikantaa” (Laaksonen, Nevasalo, Tomula 2006 s.174-175)

- ”Mahdollisuudella sopeuttaa järjestelmän ulkoasu ja muu olemus organisaation omaan ilmeeseen voidaan helpottaa muun muassa käytön oppimista” (Laaksonen, Nevasalo, Tomula 2006 s.174-175)

- ”Hankittavissa järjestelmissä on otettu huomioon lainsäädännön asettamat säädökset yksityisyydensuojan sekä tietoturvan osalta. Ulkomaisen ohjelmiston osalta voidaan todeta että tämä asia on harvoin otettu huomioon.” (Laaksonen, Nevasalo, Tomula 2006 s.174-175)

Monien tutkimusten mukaan käyttäjillä on liian monta tunnusta eri järjestelmiin muistettavinaan. Tästä syystä olisikin hyvä integroida järjestelmiä tai ottaa käyttöön järjestelmä joka käyttää vain yksiä tunnuksia käyttäjän todentamiseen ja siirtää ne sitten eteenpäin palvelulle joka vaatii käyttäjän tunnistusta. Tämänlainen ratkaisu säästäisi paljon aikaa ja vaivaa sen käyttäjiltä sekä ylläpitäjiltä. (Laaksonen,Nevasalo,Tomula. 2006)

5 KÄYTTÄJÄTUNNISTUS TEORIASSA

Käyttäjätunnistuksen perusjuoneen kuuluu se että tunnistetaan juuri se käyttäjä, joka käyttäjä väittää olevansa. Käyttäjätunnistus voi tapahtua kolmella keinolla tai sitten niiden yhteisellä menetelmällä. Yksi tunnistautumismenetelmä on kysyä jotain mitä käyttäjä tietää, kuten salasana tai pinkoodi. Toinen metodi on pyytää käyttäjältä jotain sellaista mitä omistaa, esim. avaimet. Kolmas keino on tunnistaa käyttäjä fyysisen ominaisuuksien mukaan. (Todorov 2007)

Nykypäivänä ihmisten käytössä on lukuisia palveluita ja oppimisalustoja jotka vaativat sisäänkirjautumisen ennen kuin kyseistä palvelua voi käyttää. Teknologia tuo tullessaan uusia tunnistautumismahdollisuuksia kuten sormenjälkitunnistimet ja biotunnisteet. Perinteisesti käyttäjän tunnistus kuitenkin tapahtuu käyttäjäntunnuksen avulla sekä todentaminen salasanalla.

Vaikka salasana ei ehkä olekaan enää se paras tapa käyttäjän todentamiseen, on se edelleen yleisin ja helpoin tapa rakentaa todennus. (Todorov 2007)

5.1 Todentaminen, Auktorisointi ja Valvonta (engl. authentication, authorization & accounting)

Kun suojataan tietojärjestelmiä joihin kirjaudutaan sisälle tai ne pitävät sisällään käyttäjätietoja henkilöistä, on tärkeätä että tieto ja järjestelmä itsessään on hyvin suojattu ja turvallisuuskriteerit ovat oikein asetettu. Yleensä järjestelmän turvallisuudessa noudatetaan kolmea yleisprosessia jotka yhdessä muodostavat toimivan ja turvallisen kokonaisuuden. Nämä kolme prosessia ovat:

- Todentaminen: yleensä kyseisellä sanalla viitataan tunnistamiseen ja todentamiseen, määritetään ja laillistetaan käyttäjän henkilöllisyys ja aitous.
- Auktorisointi: Tarjotaan pääsy niihin tietoihin ja palveluihin niille käyttäjille joilla on siihen oikeus, ja taas toisinpäin estetään pääsy käyttäjiltä niihin resursseihin joihin heillä ei ole oikeutta.
- Valvonta: tarjoaa mahdollisuuden seurata käyttäjien jälkiä, mitä he järjestelmissä tekevät ja mitä resursseja he käyttävät. (Todorov 2007)

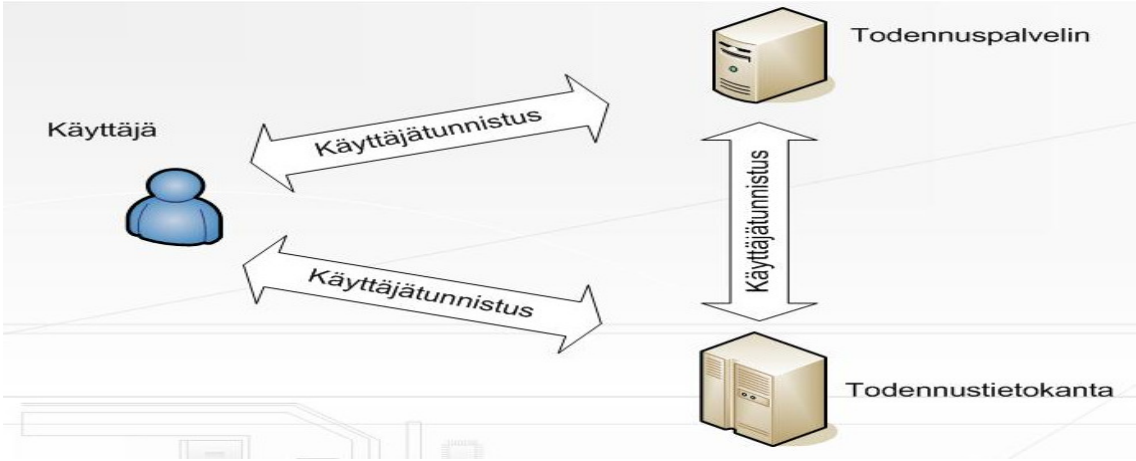
5.2 Tunnistaminen ja autentikointi (Identification & Authentication)

Autentikointi (engl. authentication) -prosessi koostuu yleisesti katsottuna kahdesta eri vaiheesta. Vaiheet ovat tunnistautuminen ja todennus. Tunnistautuminen (engl. identification) tarjoaa palvelulle käyttäjätunnistatumisen joka yleensä tapahtuu userID:n, eli käyttäjätunnuksen avulla. Palvelu jonne kirjaudutaan, tarkistaa löytyykö kyseinen käyttäjä tietokannasta ja mitkä ovat hänen oikeutensa kyseiseen järjestelmään. Tämä tapahtuu tarkastelemalla käyttäjätiedoista eri objekteja joita hänelle on asetettu. Kaiken tämän jälkeen tunnistaminen on suoritettu ja siirrytään seuraavaan vaiheeseen. (Todorov 2007)

Autentikointivaiheessa todennetaan käyttäjä. Vaikkakin käyttäjän antama käyttäjätunnus löytyisi kyseisestä järjestelmästä, se ei välttämättä tarkoita sitä että käyttäjä olisi juuri se joka väittää olevansa. Varmistus käyttäjän oikeellisuudesta saadaan kun kysytään käyttäjän häneltä vaadittua todennustekniikkaa, esim. salasanaa. Todistetta jota pyydetään käyttäjältä esitettäväksi käyttäjätunnistuksen yhteydessä, kutsutaan myös varmenteeksi (credentials). Eri järjestelmät voivat vaatia eri määrän varmenteita. Tietokoneiden järjestelmissä yleensä varmenteet ovat käytössä salasanan muodossa. Salasana on salainen ja sen tietää vain käyttäjä itse ja järjestelmä. Varmenteita voi olla muitakin kuin edellä mainitut, kuten PIN-koodi tai sertifikaatti. Kun käyttäjä on tunnistettu, on hän valmis käyttämään hänelle määrättyjä re-

sursseja. Tyypillinen käyttäjäntunnistusprosessi tapahtuu kolmen komponentin avulla (KUVIO 6). Komponentit ovat:

- Käyttäjä: osapuoli käyttäjätunnistuksessa joka tarjoaa identiteetin sekä todisteen identiteetilleen. (Todorov 2007)
- Autentikaattori (engl. authenticator): tarjoaa resursseja käyttäjälle ja varmistaa käyttäjän identiteetin valtuuttaakseen käyttäjän pääsyn eri palveluihin ja resursseihin. (Todorov 2007)
- Käyttäjätietokanta, Varasto tai mekanismi joka tarkistaa käyttäjän varmenteen. Tämä voi olla hyvinkin yksinkertainen järjestelmä tai palvelin joka toimii verkossa ja tarjoaa keskitettyä käyttäjätunnistusta organisaatiolle tai internettiin. (Todorov 2007)



Kuvio 6: Käyttäjätunnistuksen komponentit. (Todorov 2007)

Käyttäjätunnistusjärjestelmän rakennetta ei ole rajattu millään tavalla. Käyttäjä, todennuspalvelin ja todennustietokanta voivat sijaita samalla koneella. Yleisesti katsottuna käyttäjä/client sijaitsevat yhdellä koneella ja palvelin ja tietokanta toisella koneella. Järjestelmän eri osapuolet voivat kommunikoida keskenään toisistaan riippumatta niille ennalta määritetyillä kielillä ja protokollilla. Esimerkiksi autentikointiprotokollista Kerberos tyypillisesti vaatii suoran kommuninkointimahdollisuuden käyttäjän ja palvelimen välillä sekä käyttäjän ja todennustietokannan välillä. Käyttäjätunnistusta katsoen palvelimen ja todennustietokannan välillä ei ole suoraa kommunikointia, mutta kuitenkin viestit käyttäjältä tietokantaan pitävät sisällään tietoa jonka palvelin on lähettänyt todennustietokantaan. (Todorov 2007)

5.3 Auktorisointi (engl. authorization)

Auktorisointi on käyttäjätunnistuksen osaprosessi jonka avulla määritetään jo tunnistautuneelle ja todennetulle käyttäjälle oikeudet eri resursseihin ja palveluihin. Yleensä auktorisoinnin hoitaa palvelua tarjoava resurssi tai järjestelmä. Tämä tarkoittaa sitä että kun esimerkiksi käyttäjä koettaa käyttää tiedostoja jotka sijaitsevat palvelimella, on palvelimen tehtävä määrätä kenellä on oikeus käyttää tiedostoja. Auktorisointi mahdollistaa tarkan kontrolloinnin. Se erottaa toisistaan eri operaatiot kuten tiedoston lukemisen, siihen kirjoittamisen, sen tuhoamisen tai esim. sovelluksen käynnistämisen. Ennen kuin auktorisointi astuu mukaan kuvioihin, on tehtävä käyttäjän tunnistaminen sekä todentaminen. Käyttäjien auktorisointi nojaa täydellisesti käyttäjien tunnistautumistietoihin, joiden avulla määritetään kykeneekö kyseinen käyttäjä palvelua käyttämään. Käyttöjärjestelmissä käyttäjätunnistusta on helpotettu erilaisilla tunnistustyökaluilla joita ohjelmistot sitten hyödyntävät. Tästä esimerkkinä Security Kernel. (Todorov 2007)

Käyttäjä voidaan autentikoida käyttämällä tiettyä identiteettiä, mutta häneltä voidaan pyytää oikeuksia jotta hän pääsee käsiksi tiedostoihin jotka sijaitsevat eri käyttäjän resurssi- tai palvelukannassa. Käyttäjän selkeästi pyytäessä lupaa ohjelmaan tai palveluun edellä mainitulla keinolla, käytetään siitä nimitystä auktorisoitu identieetti (authorization identity). Kun sama operaatio tapahtuu palvelun tai ohjelman kautta tunnetaan se myös toisena esiintymisenä (impersonation). Impersonaation tapauksessa käyttäjä voi pitää hallussaan tunnistautumisidentiteetin joka on varmistettu käyttäjätunnistusprosessissa. Impersonaatio on erittäin kätevä menetelmä asiakas/palvelin ympäristössä. Käyttäjille jotka ottavat yhteyden palvelinohjelmistoon voidaan tarjota pääsy resursseihin ilman että käyttäjien sitä tarvitsee erikseen tehdä, eli prosessi tapahtuu ohjelman kautta. Se antaa myös mahdollisuuden ottaa yhteyden palvelimeen käyttäen jonkun toisen laajempaa tai enemmän rajattua pääsylupaa. (Todorov. 2007)

5.4 Käyttäjän kirjautumisprosessi

Autentikointi ja auktorisointi kulkevat hyvin pitkälti käsi kädessä. Onkin vaikea erotella missä vaiheessa autentikointi loppuu ja missä kohtaa auktorisointi alkaa. Teoriassa autentikoinnin tehtävä on vain varmistaa käyttäjän identiteetti, kun taas auktorisointi on vain vastuussa siitä onko kyseisellä käyttäjällä oikeus tiettyyn resurssiin tai palveluun. (Todorov 2007)

Tarjotakseen keskenäisen riippuvuuden autentikaation ja autorisaation välille käyttöjärjestelmät ja ohjelmat tarjoavat yleensä työkalun tai apuvälineen niin sanottuun käyttäjän kirjautumisprosessiin. Nämä apuvälineet tarjoavat käyttäjälle mahdollisuuden tunnistautua sekä aloittavat viestin vaihdon käyttäjän ja järjestelmän välillä. Nämä käyttäjätunnistustyökalut

tarjoavat myös käyttäjälle ohjelma- tai järjestelmäkohtaisen pääsyvaltuuden (access token). (Todorov 2007)

Kun käyttäjä käynnistää ohjelman tai kirjautuu johonkin järjestelmään, on hänellä käytössään access token joka pitää sisällään tietoa käyttäjän identiteetistä sekä siitä onko kyseisellä henkilöllä oikeutta päästä käsiksi kyseiseen resurssiin. Access token sijoittuu käyttäjäntunnistusprosessissa tunnistautumisen ja auktorisoimisen välille. Kirjautumisprosessi voi myös tehdä asioita jotka eivät liity turvallisuuteen; esimerkiksi käyttäjän kirjautuessa järjestelmään, järjestelmä hakee automaattisesti oikeat säädöt ja asetukset käyttäjälle. (Todorov. 2007)

5.5 Valvonta (engl. Accounting)

Käyttäjät ovat itse vastuullisia omista tekemisistään järjestelmien sisällä. Käyttäjät voidaan oikeuttaa pääsemään tiettyihin resursseihin ja kun he käyttävät niitä, järjestelmien ja ohjelmien täytyy tarjota mahdollisuus seurata tätä kaikkea. Mahdollisista väärinkäytösyrityksistä pitää jäädä seurattava jälki; kuka järjestelmään on yrittänyt kirjautua ja mihin resurssiin. (Todorov 2007)

Accounting eli valvonta on prosessi jonka tehtävä on seurata järjestelmissä ja palveluissa tapahtuvia käyttäjien jättämiä jälkiä. Valvontaprosessi on erittäin hyvä asia tietoturvallisuuden kannalta, sillä sen avulla väärinkäytökset voidaan huomata jos niitä esiintyy. Se myös pitää kirjaa kirjautumisista, onnistuivat ne tai ei. Käyttäjä ei välttämättä jostain syystä pääse aina kirjautumaan järjestelmään ja tämä pitää olla seurattavissa. (Todorov 2007)

6 HAKA-Luottamusverkosto

Haka on korkeakoulujen sekä tutkimuslaitosten yhteinen käyttäjätunnistusjärjestelmä. Hakan käyttäjätunnistautumisjärjestelmä perustuu sen luomaan omaan luottamusverkostoon eli federaatioon. Siihen liitytään kirjoittamalla palvelusopimus CSC – Tieteen Tietotekniikan keskus Oy:n kanssa joka toimii samalla luottamusverkostoon operaattorina. Verkoston jäseneksi voivat liittyä niin yliopistot, ammattikorkeakoulut, yliopistosairaalat, tutkimuslaitokset, kuin näille tukea antavat yrityksetkin. Luottamusverkostoon voivat myös liittyä ne tahot jotka tarjoavat joitain palveluita organisaatioille. Hakan tarkoituksena on tarjota organisaation rajojen yli kirjautuminen. Itse toteutus tapahtuu Shibboleth-järjestelmällä. (CSC Tietotekniikan keskus 2009)

7 shibboleth

Shibboleth on standardeihin perustuva avoimen lähdekoodin ohjelmisto joka mahdollistaa käyttäjien lähettää yksityistä tietoa itsestään etäpalveluihin. Tätä tietoa voidaan käyttää

valtuuttamiseen, autentikointiin, sisällön personalisoimiseen sekä tunnistautumiseen eri palveluissa. Sen on kehittänyt internetissä toimiva internet2-yhteisö. Shibbolethissa on kaksi pääkomponenttia, Identity Provider sekä Service Provider. Identity Provider kerää tietoa käyttäjästä valituista lähteistä, valmistelee sen ja lähettää Service Providerille. Service Provider valmistelee saadun tiedon ja käyttää sitä sisältöjen suojaamiseen sekä mahdollisesti antaa sen sovellusten käyttöön.

Shibboleth toimii federoituna järjestelmänä joka mahdollistaa tietoturvallisen pääsyn eri palveluihin ja resursseihin eri tietoverkkojen sisällä. Shibboleth-federaatiota ei määritellä järjestelmässä niinkään teknisesti, vaan se muodollisesti mahdollistaa samaan federaatioon kuuluvien Providerien luottaa toisiinsa skaalautuvasti.

Shibboleth käyttää referenssinä kommunikointiinsa Identity Providerien ja Service Providerien kesken metadataa. Metadata on tiedosto Shibboleth-järjestelmässä, jossa määritellään mistä mikäkin Provider löytyy ja mitä ominaisuuksia se omaa. Federaatioilla on yleensä käytössä yksi metadata-tiedosto johon on määritetty federaatioon kuuluvien Providerien ominaisuudet. Uuden Providerin liittyessä federaatioon sen tiedot täytyy sisällyttää metadataan ja jakaa metadata federaation jäsenille. Yleensä federaatioon kuuluvat Providerit konfiguroidaan hakemaan metadata-tiedosto verkon kautta federaation ylläpitäjältä tietyin väliajoin, jotta koko federaatio pysyy päivitettynä jäsenistään.

Tyypillisessä Shibboleth-skenaariossa käyttäjä avaa suojattua palvelua internet-selaimella, siirtyy tunnistautumaan Identity Providerille ja päätyy Service providerin kautta suojattuun palveluun tunnistautumisen onnistuessa. Tämän operaation voi jakaa karkeasti neljään vaiheeseen:

Vaihe 1: käyttäjä avaa suojatun resurssin

Käyttäjä yrittää avata suojattua resurssia jonka tuloksena Service Providerin Shibboleth daemon käynnistyy ja ottaa tapahtuman haltuunsa. Service Provider katsoo konfiguiroinneistaan mihin Identity Provideriin se ottaa yhteyden tunnistautumista varten ja mitä protokollia yhteydessä käytetään. Konfiguroinneista riippuen se voi avata tekstisyöttöikkunan selaimeen jossa määritetään Identity Provider, lähettää käyttäjän Discovery Serviceen valitsemaan Identity Provider tai lähettää käyttäjän jo valmiiksi määritettyyn Identity Provideriin. (Internet2 2009)

Vaihe 2 : käyttäjä tunnistautuu Identity Providerille

Käyttäjä päätyy valitsemalleen tai konfigurointien mukaiselle Identity Providerille. Identity Provider päättää konfigurointiensa mukaan mitä tunnistautumismetodia se käyttää. Se voi avata käyttäjälle sisäänkirjautumisikkunan tai esim. tunnistaa tämän IP-osoitteen perusteella. Käyttäjä tunnistautuu Identity Providerin valitsemalla metodilla. (Internet2 2009)

Vaihe 3: Identity Provider antaa vastauksen Service Providerille

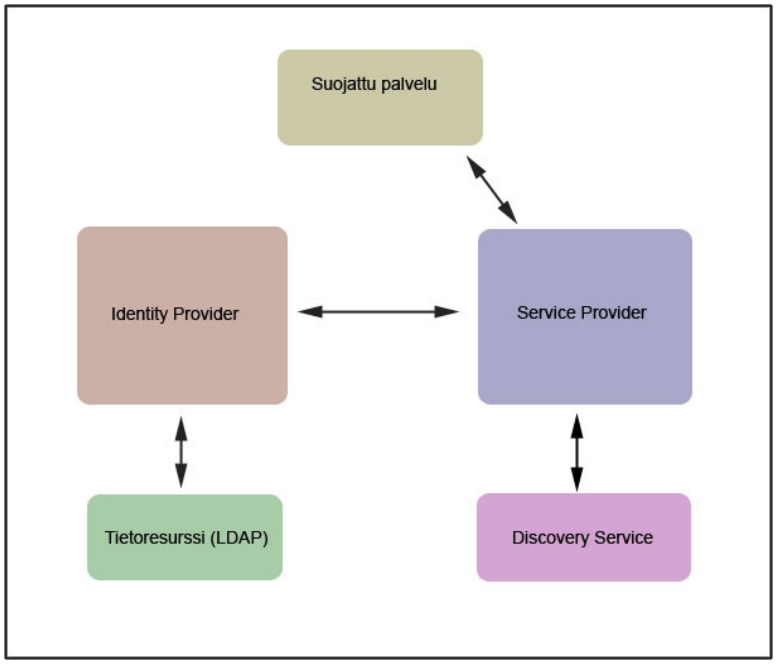
Identity Provider kerää tunnistautuneen käyttäjän attribuutit tietoresursseistaan, muuntaa ne tarvittaessa oikeaan muotoon ja liittää koodekit jokaiseen saatuun attribuuttiin. Tämän jälkeen se tarkistaa konfiguroinneistaan mitä attribuutteja se saa lähettää eteenpäin kyseessä olevalle Service Providerille. Käyttäjästä lähetettävät tiedot pakataan viestiin aiemmin liitettyjen koodekkien kanssa. Viesti allekirjoitetaan Identity Providerin salausavaimella ja salataan Service Providerin salausavaimella tietoturvan varmistamiseksi. Tämän jälkeen käyttäjä ohjataan takaisin Service providerille pakatun viestin kanssa. (Internet2 2009)

Vaihe 4: Takaisin Service Providerilla

Service Provider ottaa vastaan pakatun viestin, purkaa sen, avaa salauksen ja tekee useita turvallisuustarkastuksia. Jos kaikki on kunnossa, se ottaa attribuutit ja muun käyttäjän tiedon viestistä. Attribuutit käännetään paikalliseen ympäristöön tai niistä tehdään muuttujia joita suojattu resurssi voi käyttää. Service Provider voi konfigurointiensa mukaan tämän jälkeen käyttää attribuutteja kulunvalvontaan tai antaa ne suojatun resurssin käyttöön. Jos attribuuttien mukainen kulunvalvonta ei estä käyttäjää, hän pääsee sisään suojattuun resurssiin. (Internet2 2009)

7.1 Järjestelmän komponentit

Shibboleth-järjestelmään kuuluu kaksi itse Shibbolethin komponenttia, Identity Provider sekä Service Provider. Järjestelmään liitetään myös sisältö jota suojataan sekä resurssit joista haetaan käyttäjätietoa. Yleensä sisältö on tietty WWW-palvelu, esim. korkeakoulun oppimisportaali ja tietoresurssi esim. LDAP-hakemisto. Järjestelmän osana voi olla myös Discovery Service jossa valitaan mihin Identity Provideriin halutaan tunnistautua.

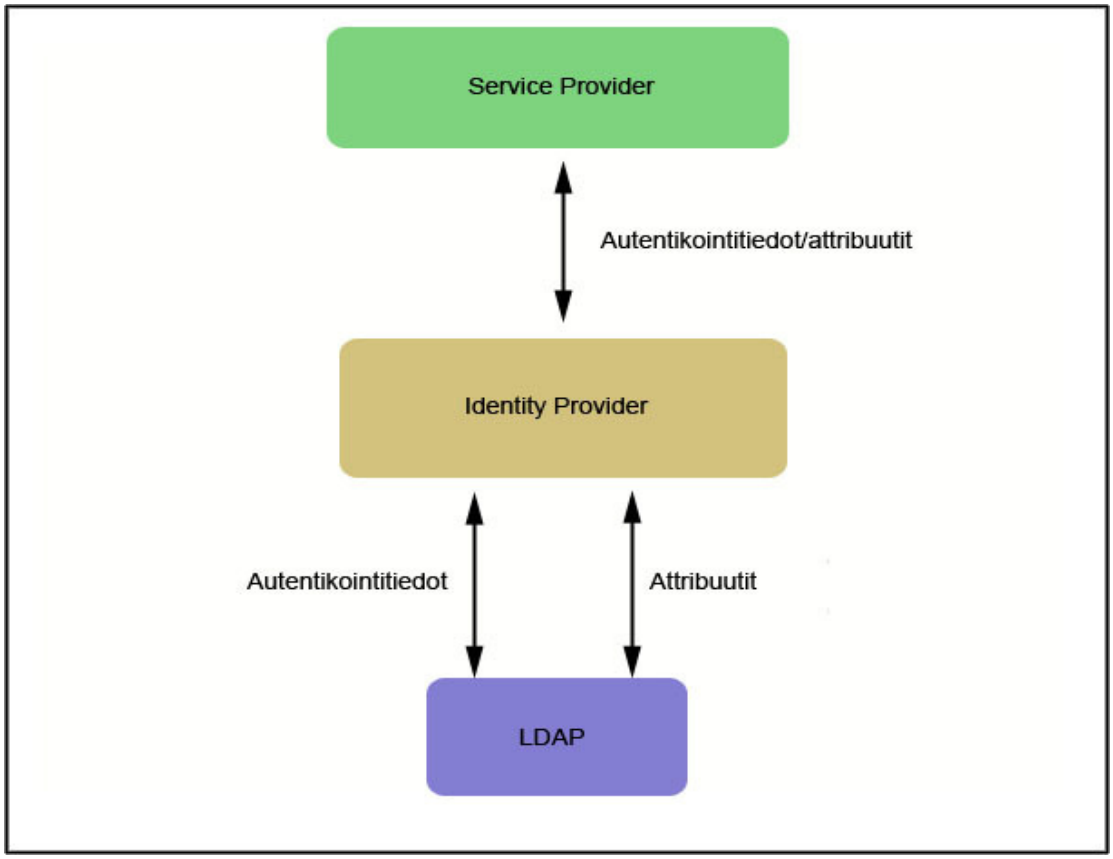


Kuvio 7: Shibboleth järjestelmän komponentit

7.1.1 Identity Provider

Identity Provider eli lyhennettynä IdP on toinen järjestelmän pääkomponenteista. Identity Provider tarjoaa nimensä mukaisesti halutun käyttäjän identiteetin ja tarvittavan tekniikan sen tietoturvalliseen hankkimiseen sekä eteenpäin lähettämiseen. IdP asennetaan fyysisesti esimerkiksi korkeakoulun serverille josta se ottaa yhteyttä korkeakoulun tietoresursseihin ja josta se voi tarjota Service Providereille tietoa korkeakoulun organisaation henkilöistä. Kaikki Idp:lle ja sieltä poispäin liikkuva tieto salataan SSL-avaimilla jotta tiedonsiirto on tietoturvallista.

Käyttäjän tunnistuksen lisäksi Idp tarjoaa myös käyttäjään liittyviä attribuutteja jotka on annettu käytettävälle tietoresurssille. Yleisin käytettävä tietoresurssi on LDAP, eli Lightweight Directory Access Protocol. LDAP sisältää käyttäjien käyttäjätunnukset ja salasanat joiden pohjalta IdP voi toteuttaa käyttäjän tunnistuksen. LDAP-hakemisto sisältää myös käyttäjien attribuutteja, kuten sähköposti, työnimike tai puhelinnumero. Attribuutit otetaan vastaan IdP:llä sille konfiguroidulla tavalla, filteröidään ja lähetetään eteenpäin Service Providerille.



Kuvio 8: Shibboleth Identity Provider

Identity Provider-sovellus on pohjimmiltaan Javapohjainen servlet jota ajetaan servlet containerilla kuten Apache Tomcat. Idp vaatii lisäksi toimiakseen riittävät Java-kirjastot ja -tietokannat. Sovellusta konfiguroidaan konfigurointitiedostoista eli käsin muuttamalla tekstiä tekstieditorissa. Identity Providerin tärkeimmät konfigurointitiedostot ovat:

- Relying-party.xml: määrittää miten Identity Provider käyttäytyy eri Service Providereita kohtaan.

- Handler.xml: Määrittää käytettävän tunnistusmetodin. Nämä metodit ovat käyttäjätunnus/salasana, IP-osoite, remote user sekä previous session.

- Login.config: Käyttäjätunnus/salasana metodissa käytetyn JAAS-autentikointimoduulin konfigurointitiedosto. Määrittää mihin LDAP-hakemistoon Idp ottaa yhteyden.

- Attribute-resolver.xml: määrittää mitä attribuutteja IdP hakee ja mistä.

- Attribute-filter.xml: määrittää mitä attribuutteja idP saa lähettää eteenpäin ja kenelle.

- Logging.xml: määrittää miten ja mitä IdP tallentaa toiminnastaan logitiedostoihinsa.

Vaikka käyttäjätunnus/salasana-autentikointimetodi LDAP-hakemistoa vastaan on yleisin, IdP tarjoaa myös muita vaihtoehtoja käyttäjän autentikointiin. Muihin vaihtoehtoihin kuuluvat:

- Remote User: antaa käytettävän servlet containerin, kuten Apache Tomcatin hoitaa käyttäjän sisäänpääsyn rajoittamisen.

- IP-address: autentikoi käyttäjän IP-osoitteen perusteella.

- Previous Session: autentikoi käyttäjän jos valmis Shibboleth-session on jo olemassa.

Näitä autentikointimetodeja voidaan käyttää kytköksissä toisiinsa. Yleensä päätunnistusmetodi on kuitenkin käyttäjätunnus/salasana jonka kautta saadaan myös käyttäjän attribuutit esim. LDAP-hakemistosta. Jokainen käytettävä autentikointimetodi määritetään omaan Loginhandleriin. Loginhandler määrittää myös autentikointimetodin ajanjakson jona se on voimassa.

Käyttäjän tunnistautuessa jollain edellä mainitulla autentikointimetodilla, hänelle luodaan Shibbolethissa session. Session sisältää tietoa siitä mitkä autentikointimetodit ovat voimassa ja miten kauan, sekä mihin suojattuihin palveluihin käyttäjä on autentikoitunut. Session sisältää myös toimettomuus aikakatkaisun, eli jos käyttäjä ei tee Shibbolethin sisällä mitään määritettyyn aikaan, session lakkaa olemasta. Sessionin aikakatkaisu menee hierarkiassa autentikointimetodin voimassaoloajanjakson ohi, joten vaikka autentikointimetodilla olisi vielä voimassaoloaikaa ja session häviää, häviää myös autentikointimetodin aktiivisuus.

Vikojen etsimiseen Identity Providerissa on logitoiminto. Logging.xml tiedostossa voidaan määrittää miten tarkasti IdP tallentaa toimintaansa logiin. IdP:n ollessa toiminnassa ja vian ilmestyessä ylläpitäjän tarvitsee vain avata logitiedosto, ja katsoa mistä vika johtuu. Tämä helpottaa huomattavasi vianetsintää.

Identity Providerin asennukseen tarvitaan Linux- tai Windows-käyttöjärjestelmäpohjainen serveri johon asennetaan IdP:n asennuspaketit, servlet container eli esim. Apache Tomcat ja Java-kirjastot ja -tietokannat. Asennuksen jälkeen IdP konfiguroidaan toimimaan halutulla tavalla. Yleisimmät konfiguroinnit jotka ylläpitäjä tekee ovat seuraavat:

- Määritetään miten IdP kommunikoi Service Providerien kanssa

- Määritetään mistä IdP hakee käytettävän metadatan

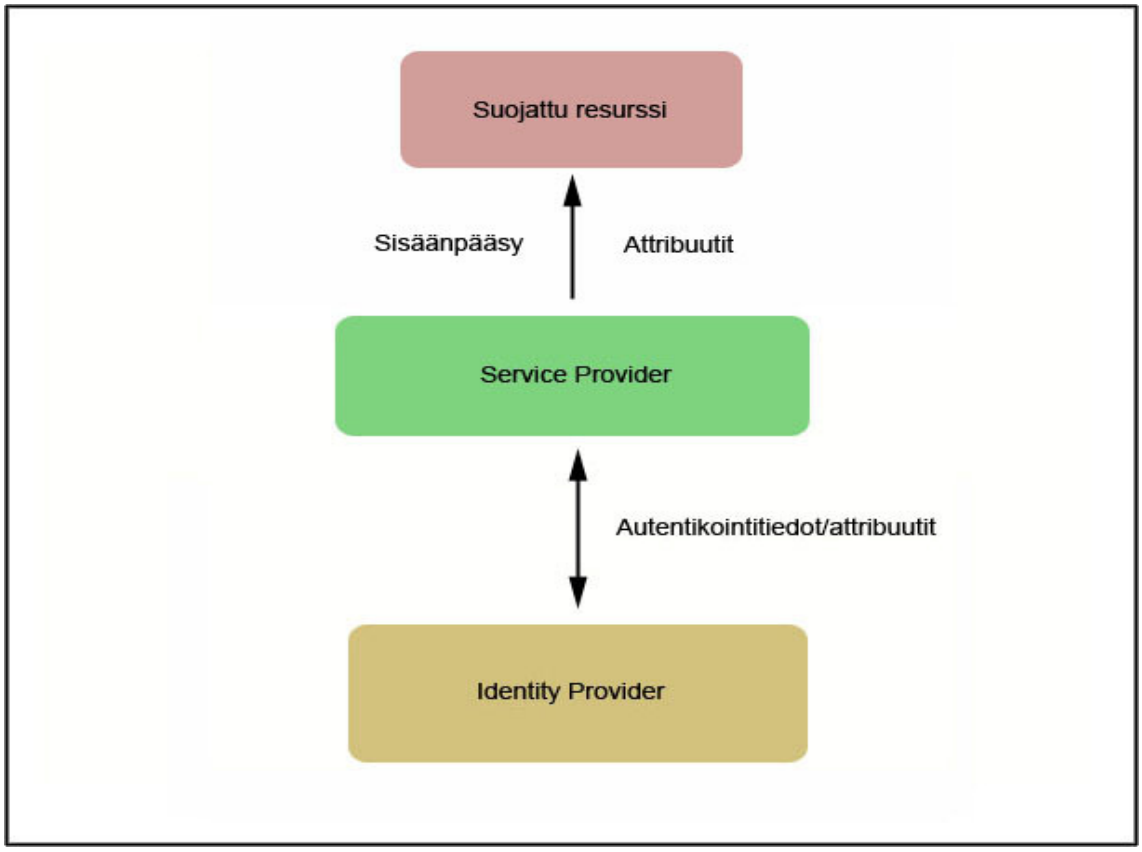
- Valitaan autentikointi/tunnistusmetodi jota käytetään
- Määritetään miten attribuutit otetaan vastaan ja miten niitä käsitellään

- Luodaan filtteri joka määrittää mitä attribuutteja saa luovuttaa millekkin Service Providerille

7.1.2 Service Provider

Service Provider, lyhennettynä SP on Shibboleth-järjestelmän komponentti joka hallitsee sisäänpääsyä sillä suojattaviin resursseihin. Resurssi on yleisimmin WWW-palvelu johon halutaan rajata pääsy. Suojatessaan palvelua/sovellusta SP lähettää käyttäjän tunnistautumaan Identity Provideriin, odottaa IdP:n vastausta ja päättää vastauksen perusteella päästääkö se käyttäjän suojattuun resurssiin. SP-ohjelmisto asennetaan fyysisesti samalle serverille missä pyöritetään suojattavaa palvelua/sovellusta.

Service Provider ottaa vastaan autentikointitietojen lisäksi myös attribuutteja. Attribuutteja voidaan käyttää kulunvalvontaan tai ne voidaan lähettää eteenpäin suojattuun resurssiin. SP voidaan esimerkiksi konfiguroida attribuuttien perusteella päästämään korkeakouluorganisaation opiskelija suojattuun resurssiin, mutta eväämään pääsy opettajalta. Attribuuteille voi olla myös käyttöä suojatussa resurssissa, niiden perusteella voidaan esimerkiksi muokata näkymää joka käyttäjälle resurssissa avautuu.



Kuvio 9: Shibboleth Service Provider

Service Provider ohjelmisto toimii kytköksissä WWW-palvelimeen, joka ylläpitää suojattua resurssia. Vain ohjelmistojen yhteistyöllä onnistuu se että WWW-palvelin voi eritellä sisällön joka suojataan Shibbolethilla ja näin lähettää käyttäjän Service Providerin kautta Identity Providerille. Suojattu resurssi määritellään virtual host ja hakemistotasolla. Virtual hostille määritellään oletusjuurihakemisto jossa sen käyttämät dokumentit/tiedostot sijaitsevat ja tämä hakemisto määritetään suojattavaksi Shibbolethilla. Näin kaikki suojattuun palveluun kuuluvat tiedostot ovat Shibbolethilla suojattuja eikä mihinkään niihin pääse käsiksi ilman että toiminta laukaisee Shibboleth-autentikoimisen käynnistymisen.

Service Providerin asennus vaatii Linux- tai Windows-pohjaisen serverin sekä SP-asennuspaketin. Lisäksi samalla serverillä tulee pyöriä WWW-palvelimella resurssi jota suojataan. Yleinen valinta WWW-palvelimeksi on Apache. SP:tä konfiguroidaan samalla tavalla kuin IdP:tä, eli muokkaamalla konfigurointitiedostoja. Service Providerin tärkeimmät konfigurointitiedostot ovat:

- Shibboleth2.xml: Service Providerin pääkonfigurointitiedosto. Tiedostossa määritellään esim. mitä palveluita suojataan ja miten Identity Providerille suuntautuva prosessi suoritetaan.

- Shibd.conf: WWW-palvelimeen liitettävä tiedosto jossa määritellään suojattava resurssi.

- Attribute-map.xml: määrittää mitä attribuutteja SP ottaa vastaan ja millä nimellä.

- Attribute.policy.xml: määrittää säännöt, miten attribuutteja käytetään.

Service Provider tekee toiminnastaan logia shibd.log tiedostoon. Loggauksen määrä ja tarkkuus määritellään Shibboleth2.xml-tiedostossa. Vianmääritystä toteutettaessa kannattaa huomioida myös WWW-palvelimen logitiedostot, koska WWW-palvelin on tietyiltä osin kytköksissä Service Provideriin.

Kun Service Provider on onnistuneesti asennettu, yleisimmät konfiguroinnit jotka ylläpitäjä tekee ovat:

- Määritetään suojattava sisältö

- Määritetään mihin Identity Provideriin tai Discovery Serviceen Service Provider ottaa yhteyksiä

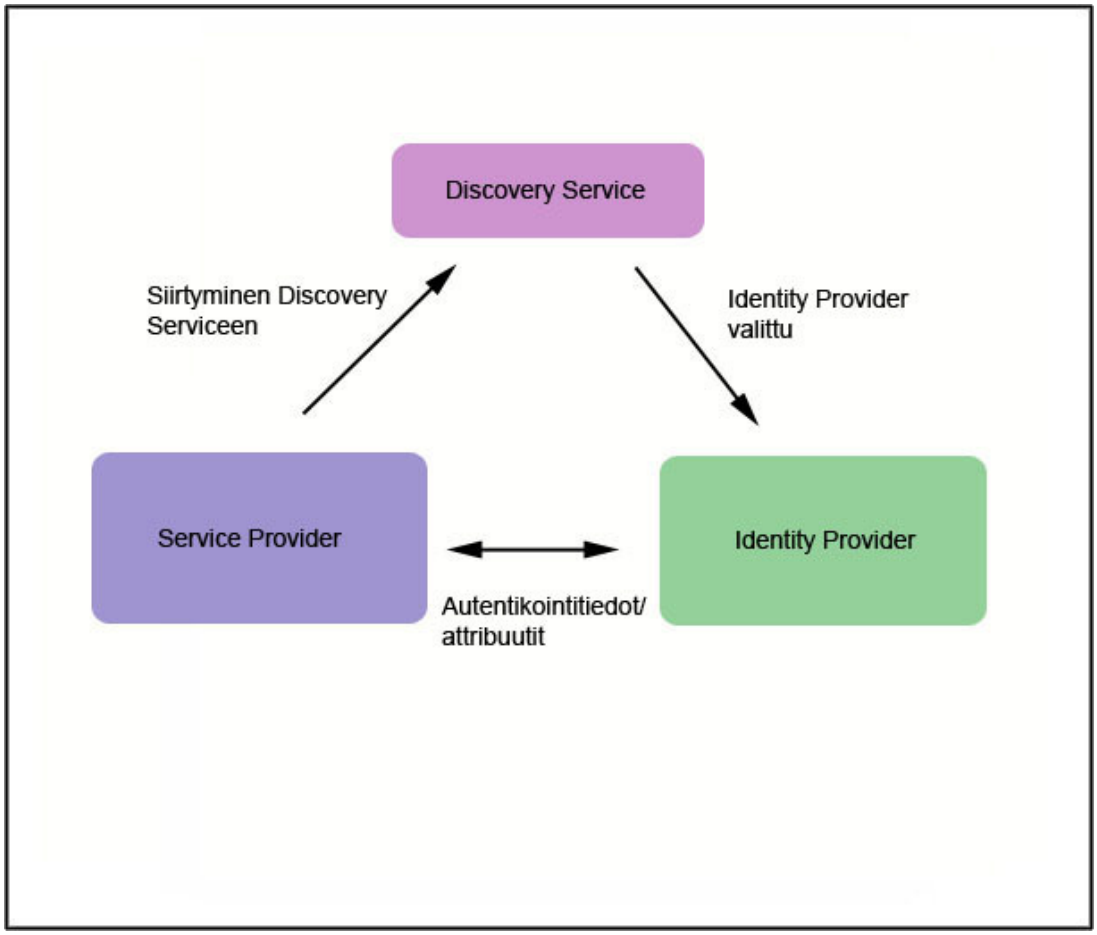
- Määritetään mistä Service Provider hakee käytettävän metadatan

- Määritetään miten Service Provider ottaa attribuutteja vastaan ja miten se käyttää niitä

- Määritetään mahdollinen kulunvalvonta suojattuun sisältöön

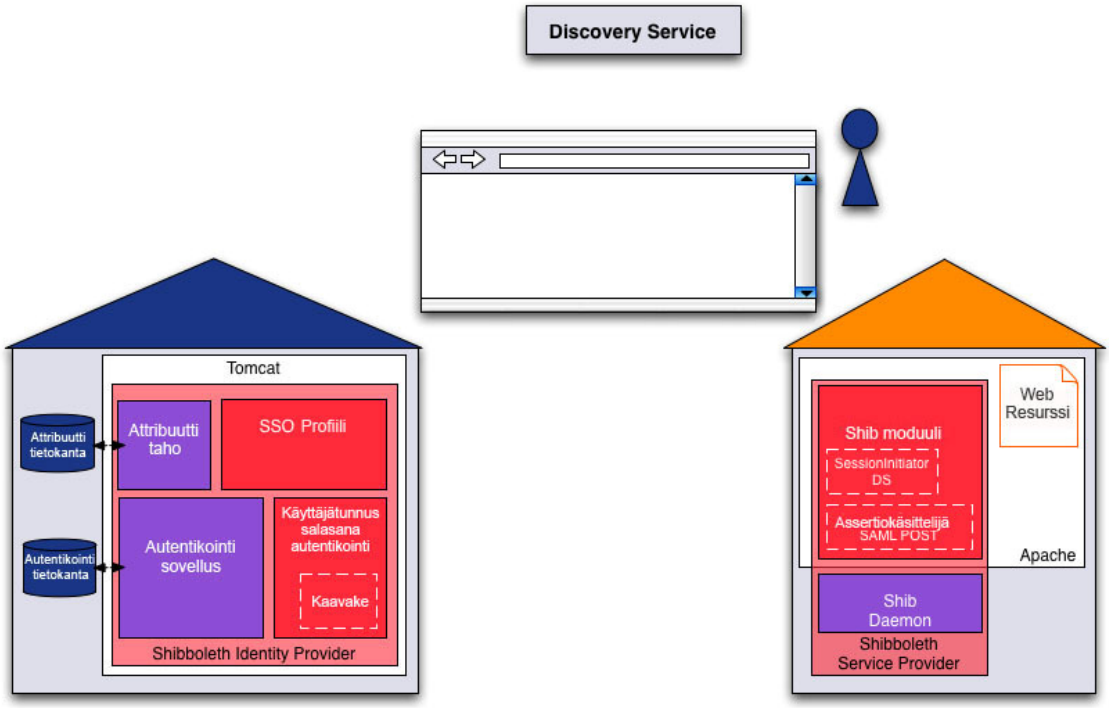
7.1.3 Discovery Service

Useimmissa tapauksissa tiettyyn Service Provideriin voi tunnistautua monen eri Identity Providerin kautta. Ongelma on miten Service Provider osaa ohjata käyttäjän oikealle Identity Providerille jossa käyttäjä pystyy tunnistautumaan. Discovery Service tarjoaa ratkaisun tähän ongelmaan. Suojattua sisältöä avatessa Service Provider konfiguroidaan ohjaamaan käyttäjä Discovery Serviceen josta käyttäjä voi valita mihin Identity Provideriin hän haluaa tunnistautua. Eli Discovery Service on Shibboleth-järjestelmän komponetti joka tarjoaa käyttäjälle mahdollisuuden valita missä hän haluaa tunnistautua, toisin sanoen mistä käyttäjän tunnukset löytyvät.



Kuvio 10: Discovery Service

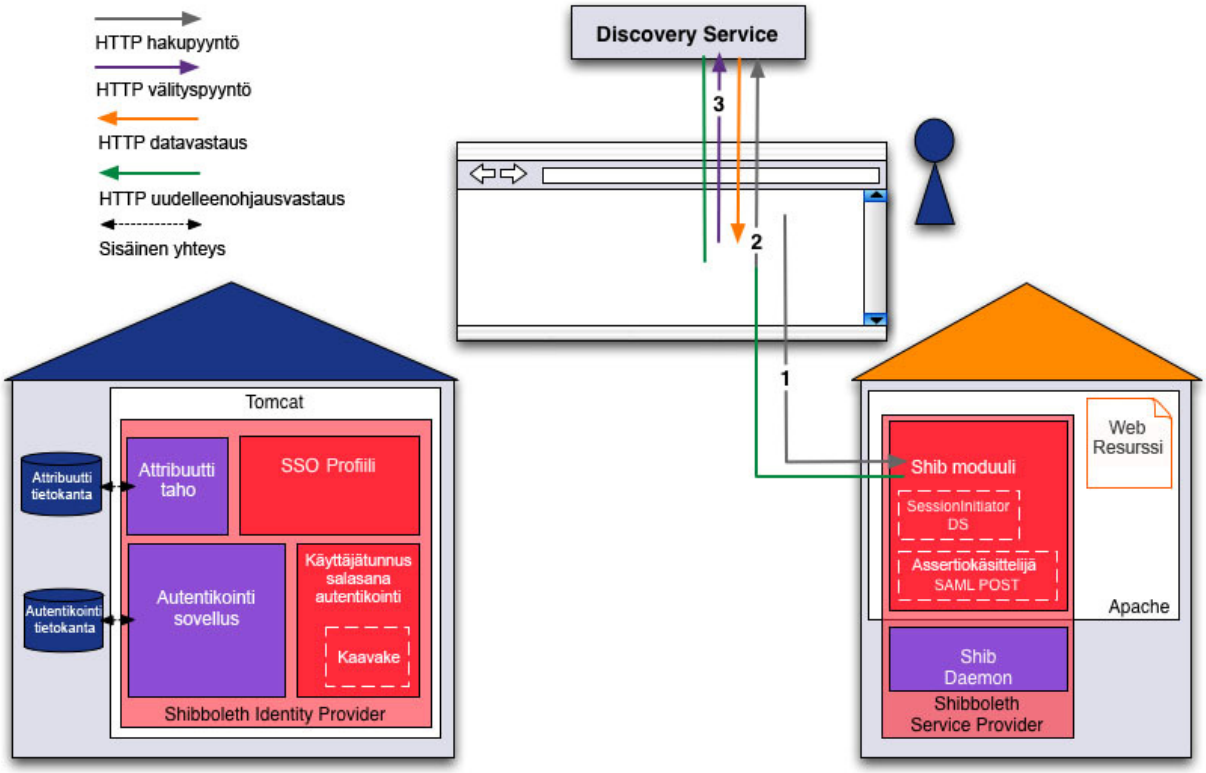
7.2 järjestelmän toiminnankuvaus



Kuvio 11: Oletustilanne. (SWITCH 2009)

Seuraavassa on eroteltu data joka lähetetään käyttäjän web-selaimesta palvelimelle (GET request ja POST request) ja data joka saadaan web-palvelimelta (redirect responses ja data responses). Joitain tietoja on korostettu käyttämällä [DETAIL POPUP] linkkiä. Tekstin lukemisen helpottamiseksi data on esitetty URL muodossa, esim. ”://” olisi '%3A%2F%2F' oikeassa muodossaan.

Vaihe 1 – Otetaan yhteys haluttuun service provideriin ja identity provideriin DS:ssän kautta.



Kuvio 12: Discovery Service. (SWITCH 2009)

1. (Käyttäjä – Selain – Service Provider)

Käyttäjä avaa web-selaimen ja ottaa yhteyttä Service Provideriin jonka osoite on https://sp.example.com/secure. Käyttäjän selain lähettää seuraavan pyynnön (KUVIO 12):

GET https://sp.example.com/secure

Koska edellä mainittu osoite on suojattu Shibboleth Service Providerilla, järjestelmä tarkistaa, että onko käyttäjä tunnistettu jo aikaisemmin ja onko hänellä olemassa oleva Shibboleth sessio. Jos käyttäjää ei tunnisteta, web-palvelin vastaa pyyntöön http-uudelleenohjauksella Discovery Serviceen joka sijaitsee osoitteessa wayf-test.switch.ch. DS-palvelimen täytyy tietää minne palauttaa käyttäjän koti organisaation valinnan, relevantti tieto saadaan seuraavasti (KUVIO 12):

302 FOUND (REDIRECT)
Set-Cookie: _shibstate_64656661756c7468747470733a2f2...
value=https://sp.example.com/secure
path=/
Location: https://wayf.example.com/WAYF
?entityID=https://sp.example.com/shibboleth

&return=https://sp.example.com/Shibboleth.sso/DS?SAMLDS=1&target=cookie

2. (Selain – DS)

Web-selain lähettää uuden pyynnön Discovery Serviceen (KUVIO 12).

GET https://wayf.example.com/WAYF
?entityID=https://sp.example.com/shibboleth
&return=https://sp.example.com/Shibboleth.sso/DS?SAMLDS=1&target=cookie

DS vastaa pyyntöön web-sivulla josta käyttäjä valitsee Identity Providerin (KUVIO 12):

200 OK
[WAYF DROPDOWN HTML PAGE]

3. (käyttäjä – selain – DS)

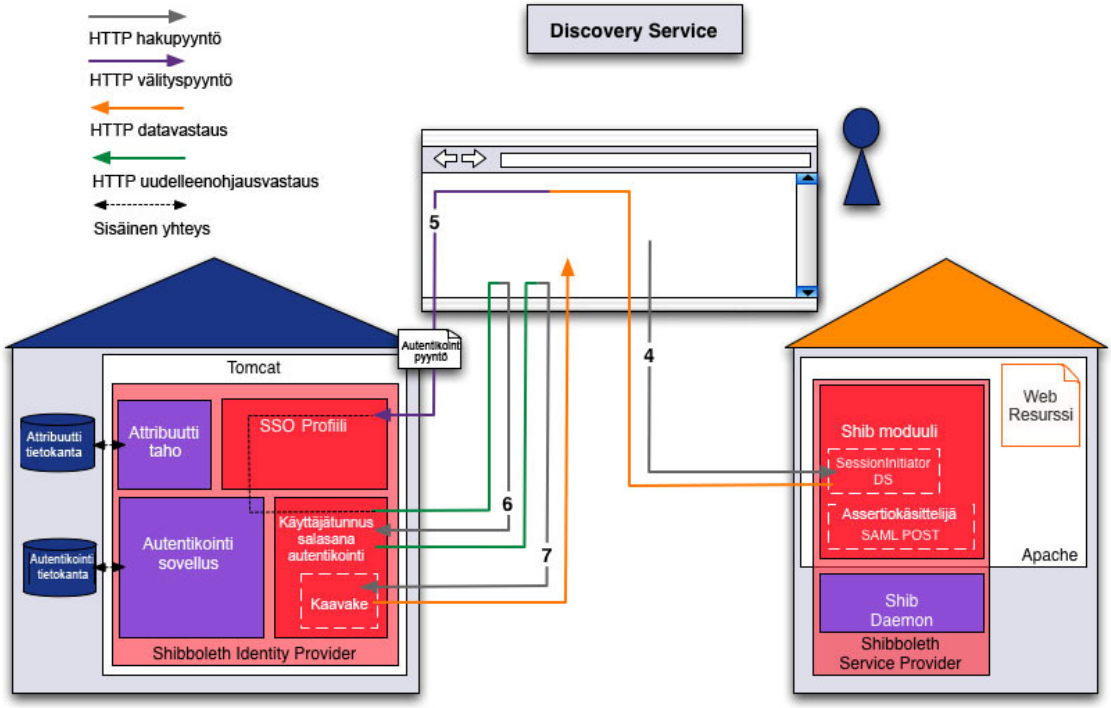
Discovery Service sivulla, käyttäjä valitsee Identity Providerin (KUVIO 12).

POST https://wayf.example.com/WAYF
POSTDATA
entityID=https://sp.example.com/shibboleth
return=https://sp.example.com/Shibboleth.sso/DS?SAMLDS=1&target=cookie
user_idp=https://idp.example.com/idp/shibboleth

DS lähettää palauteosoitteen ja idp valinnan udelleenohjauksena (KUVIO 12):

302 FOUND (REDIRECT)
Location: https://sp.example.com/Shibboleth.sso/DS?
?SAMLDS=1
&target=cookie
&entityID=https://idp.example.com/idp/shibboleth

VAIHE 2 – Session käynnistäminen ja autentikointipyyntö



Kuvio 13: Session alullepanija (engl.initiator)ja autentikaatiopyyntö. (SWITCH 2009)

4. (selain – service provider)

Viimeisen uudelleenohjauksen vastauksen johdosta selain lähettää seuraavan pyynnön (KUVIO 13):

GET https://sp.example.com/Shibboleth.sso/DS
?SAMLDS=1
&target=cookie
&entityID=https://idp.example.com/idp/shibboleth

Cookie: _shibstate_64656661756c7468747470733a2f2...
value=https://sp.example.com/secure

Session käynnistäjä tekee autentikointipyynnön ja palauttaa sen auto-submit-post-formin kanssa (KUVIO 13):

200 OK
[AUTHN REQUEST POST FORM HTML PAGE]

5. (Selain – Identity Provider)

Selain lähettää automaattisesti seuraavan pyynnön käyttäen Javascriptiä (KUVIO 13):

POST https://idp.example.com/idp/profile/SAML2/POST/SSO
POSTDATA
RelayState=cookie
SAMLRequest=PHNhbWxwOkF1dGhuUmVxdWVzdCB4bWxuczp...

Identity Provider tarkistaa autentikaatiopyynnön, koska käyttäjää ei ole vielä tunnistettu. Se lähettää uudelleenohjauksen sopivaan loginhandleriin (käyttäjätunnus/salasana)(KUVIO 9)
302 MOVED TEMPORARILY (REDIRECT)
Set-Cookie: JSESSIONID
value=C22C16A197CB9606067A1A577EF5D996
Path=/idp
Secure
Location: https://idp.example.com/idp/Authn/UserPassword

6. (selain – identity provider)

Web-selain uudelleenohjataan käyttäjätunnus/salasana-handleriin (KUVIO 13).
GET https://idp.example.com/idp/Authn/UserPassword
Cookie: JSESSIONID
value=C22C16A197CB9606067A1A577EF5D996

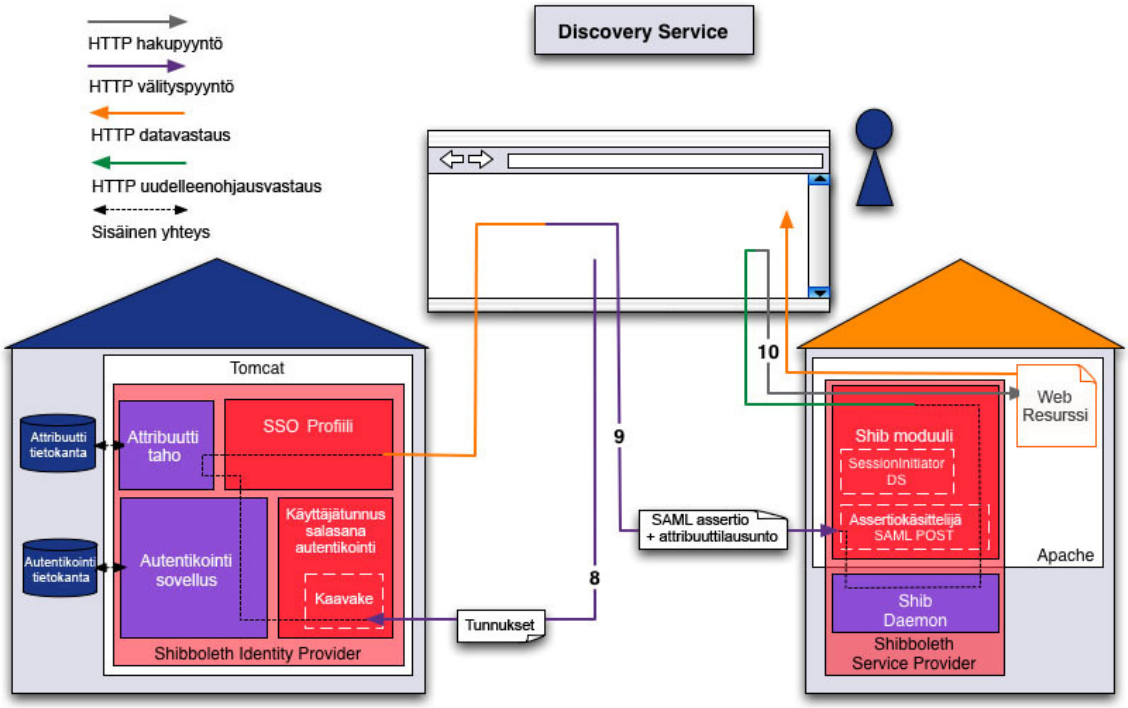
Identity Provider uudelleenohjaa tietylle käyttäjätunnus/salasana-sivulle (KUVIO 13):
302 MOVED TEMPORARILY (REDIRECT)
Location: https://idp.example.com/idp/login.jsp
?actionUrl=/idp/Authn/UserPassword

7. (selain – Identity Provider)

Jälkeenpäin selain lähettää web-sivulle GET-pyynnön käyttäjätunnukselle ja salasanalle (KUVIO 13).
GET https://idp.example.com/idp/login.jsp?
actionUrl=/idp/Authn/UserPassword
Cookie: JSESSIONID
value=C22C16A197CB9606067A1A577EF5D996

web palvelin vastaa käyttäjätunnuksella/salasana sivulla (KUVIO 13):
200 OK
[USERNAME PASSWORD LOGIN FORM HTML PAGE]

VAIHE 3 – autentikaatio, attribuuttikannanotto (engl. statement) ja pääsy



Kuvio 14: Autentikaatio, attribuuttikannanotto ja pääsy. (SWITCH 2009)

8. (käyttäjä – selain – identity provider)

Käyttäjä syöttää käyttäjätunnuksensa ja salasanansa ja lähettää ne identity providerille (KUVIO 14):

POST https://idp.example.com/idp/Authn/UserPassword
POSTDATA
j_username=demouser
j_password=demo
Cookie: JSESSIONID
value=C22C16A197CB9606067A1A577EF5D996

Identity Providerin autentikaattori varmistaa syötetyt tunnukset. Kun käyttäjän tunnukset on varmistettu onnistuneesti, attribuuttitaho suorittaa attribuuttien selvityksen ja filtteröinnin. Tämän jälkeen luodaan HTML sivu joka pitää sisällään SAML-tunnistusselosteen. Koska seloste pitää sisällään autentikaatiolausunnon lisäksi myös attribuuttilausunnon, käytetään tästä nimitystä ”attribute push”. Tunnisteseloste lähetetään käyttämällä auto-submit-post-formia (KUVIO 14):

200 OK

Set-Cookie: _idp_session
value=4m2ETlKYtvbNEmBzVNo3UHLuKSdo3HqTUqAmeZiar94=
Path=/idp
[ASSERTION POST FORM HTML PAGE]

9. (selain – service provider)

Web-selain lähettää välittömästi seuraavan pyynnön (KUVIO 14):

POST https://sp.example.com/Shibboleth.sso/SAML2/POST
POSTDATA
RelayState=cookie
SAMLResponse=PD94bWwgdmVyc2lvbj0iMS4wIiBlbmNvZGl...

Cookie: _shibstate_64656661756c7468747470733a2f2...
value=https%3A%2F%2Faai-demo.switch.ch%2Fsecure

Service Provider prosessoi SAML-tunnisteselosteen joka pitää sisällään autentikointi ja attribuuttilausunnot. Lopuksi se lähettää uudelleenohjauksen aiemmin pyydettyyn resurssiin, jonka URL on tallennettu _shibstate evästeeseen (engl.cookie) (KUVIO 14).

302 FOUND (REDIRECT)
Set-Cookie: _shibstate_64656661756c7468747470733a2f2...
value=
path=/
Set-Cookie: _shibsession_64656661756c7468747470733a2f2...
value=_0b6d4e89d2e9c4481738094f2a2c9de0
path=/
Location: https://sp.example.com/secure

10. (selain – service provider)

Sama kuin ensimmäisessä vaiheessa, selain pyytää uudelleen suojattua resurssia osoitteesta (KUVIO 14):

https://aai-demo.switch.ch/secure:

GET https://sp.example.com/secure
Cookie: _shibstate_64656661756c7468747470733a2f2...
value=
Cookie: _shibsession_64656661756c7468747470733a2f2...
value=_0b6d4e89d2e9c4481738094f2a2c9de0

Tässä tapauksessa vain käyttäjä on jo autentikoitu. Päättääkseen pääsekö käyttäjä suojattuun resurssiin, mod_shib moduuli, joka on yhdistetty Apache web-palvelimeen tarkastaa Shibbolethin pääsysäännöt ja vertaa niitä käyttäjän attribuutteihin. Tässä demossa käytetään seuraavaa pääsysääntöä (kuka tahansa auntentikoitu käyttäjä pääsee sisään)

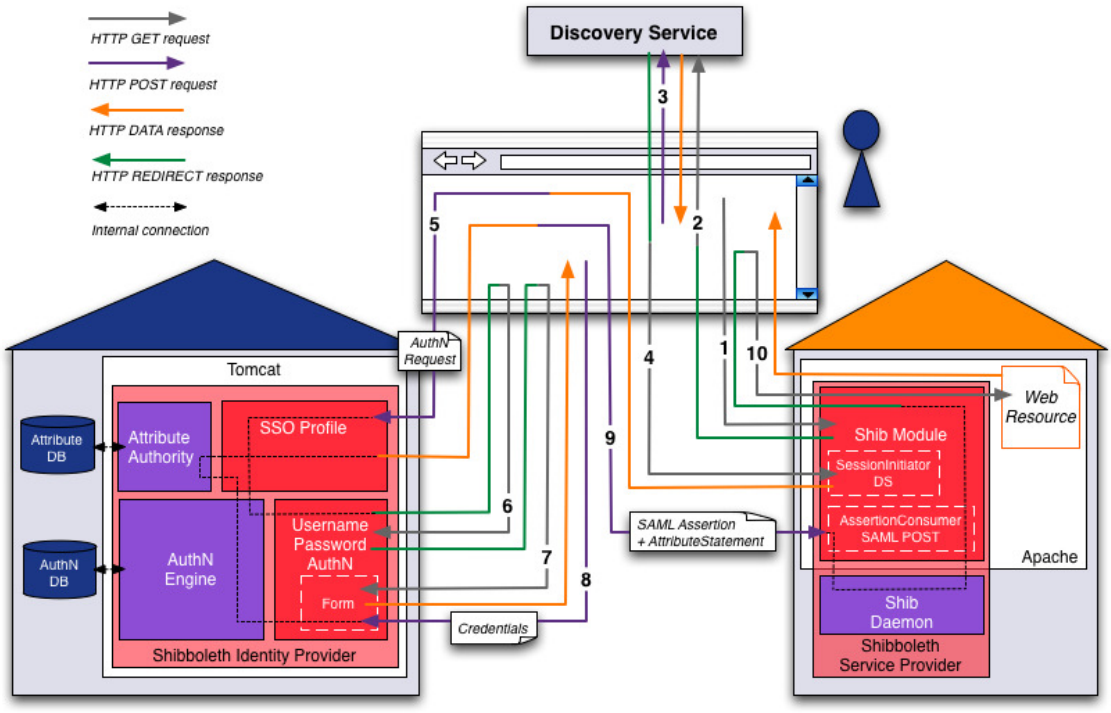
```
# content of secure/.htaccess
AuthType shibboleth
ShibRequireSession On
require valid-user
```

Service Provider palauttaa sivun ja sen sisällön (KUVIO 14):

200 OK

[RESOURCE HTML PAGE]

Yhteenveto - Shibboleth kirjautumisprosessi



Kuvio 15: Koko kirjautumisprosessi kuvattuna. (SWITCH 2009)

8 Asennus

Tässä osiossa käsittelemme tarvittavien ohjelmistojen ja järjestelmän komponenttien asennuksen.

8.1 Käyttöjärjestelmä

Sekä IdP:n että SP:n laitteistoon asennetaan käyttöjärjestelmäksi Linux CentOS 5.3.

Asennus tapahtuu boottamalla tietokone asennusDVD:ltä. Käytettävä tietokone ei välttämättä boottaa mediaa DVD:ltä, joten BIOS:n boot järjestykseen saattaa joutua tekemään tarvittavat muutokset.

Koneen käynnistyessä näkyviin tulee valintaruutu, josta valitaan enteriä painamalla graafinen asennus.

Seuraavassa ruudussa on mahdollisuus testata asennusDVD:n toimivuus valitsemalla OK. Jos testausta ei haluta suorittaa valitaan Skip. Tämän jälkeen asennusohjelma latautuu hetken aikaa, jonka jälkeen suraavassa ruudussa valitaan Next.

Seuraava ruutu antaa kielivalinnat. Tässä asennuksessa kieliksi valitaan Finnish (suomi). Näppäimistön merkistöksi valitaan myös suomalainen.

Valitaan Asenna CentOS ja painetaan Seuraava.

Kiintolevynosiointiruudussa pidetään oletusasetukset jos asennuksen kohteena olevalla koneella ei ole muuta käyttöjärjestelmää asennettuna ja varmistetaan seuraavassa ruudussa painamalla Kyllä.

Seuraavaksi määritellään koneen nimi. Tässä asennuksessa Service Provider koneelle sp.example.com ja IDP koneelle idp.example.com.

Aikavyöhykkeeksi valitaan Eurooppa/Helsinki.

Luodaan pääkäyttäjän(root) salasana ja vahvistetaan se.

Seuraavasta ruudusta ruksataan laatikot Desktop – gnome, Server ja Server GUI ja painetaan Seuraava.

Aloitetaan varsinainen asennus valitsemalla Seuraava.

Aseenuksen valmistuttua kone käynnisetään uudelleen.

Uudelleen käynnistyksen jälkeen määritellään käyttöjärjestelmän yleisesiä asetuksia:

Palomuurin asetuksiin voi valita luotetut palvelut, tässä asennuksessa palomuuri poistetaan kokonaan käytöstä valitsemalla Pois päältä. Myös SELinuxin asetukseksi valitaan Pois päältä.

Seuraavaksi voimme mahdollistaa Kdumpin käytön jolla järjestelmän ytimen kaatumisesta saa vedoksen sen mahdollisesti tapahtuessa. Tässä asennuksessa emme ota sitä käyttöön.

Asetetaan päivämäärä ja aika oikeiksi jos ne ei eivät sitä ole.

Järjestelmään voi luoda halutessaan käyttäjän. Emme kuitenkaan luo käyttäjää tässä asennuksessa vaan teemme kaikki asennukset pääkäyttäjänä.

Seuraavaksi voimme määrittää äänikortin asetuksia sekä asentaa lisäohjelmistoja erillisilta CD:ltä. Emme tee muutoksia näihin kohtiin asennuksessa. Valintojen jälkeen kone käynnistyy taas uudestaan.

Koneen käynnistyessä Käyttäjänimeksi kirjoitetaan root ja salasanaksi aiemmin luotu pääkäyttäjän salasana. CentOS 5.3 asennus on valmis.

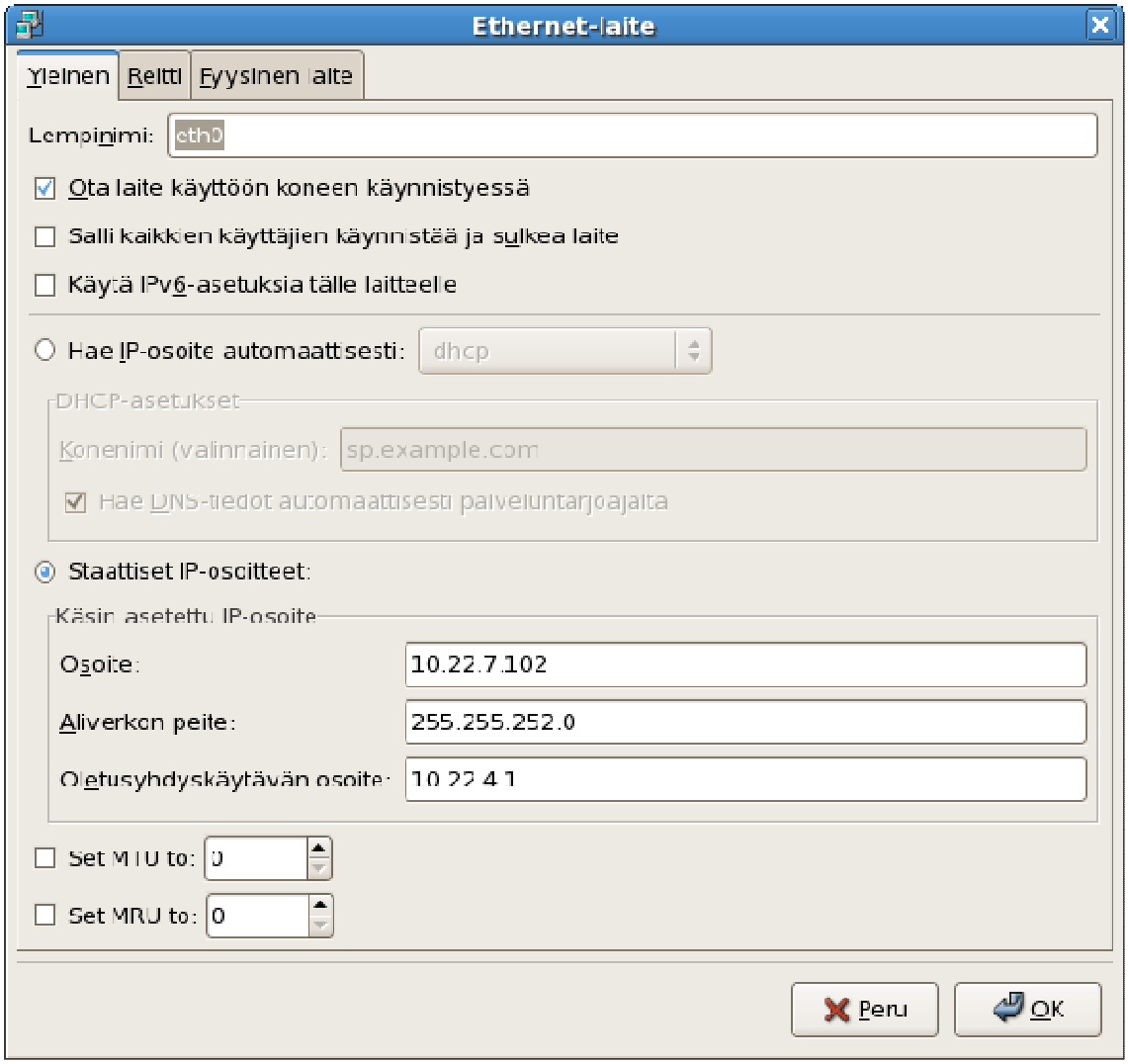
8.2 Service Provider asennus

Tässä osiossa käymme läpi Service Providerin asennuksen. Ennen Shibboleth Service Providerin asennusta tehdään muutamia esivalmisteluja.

8.2.1 Verkkoasetukset

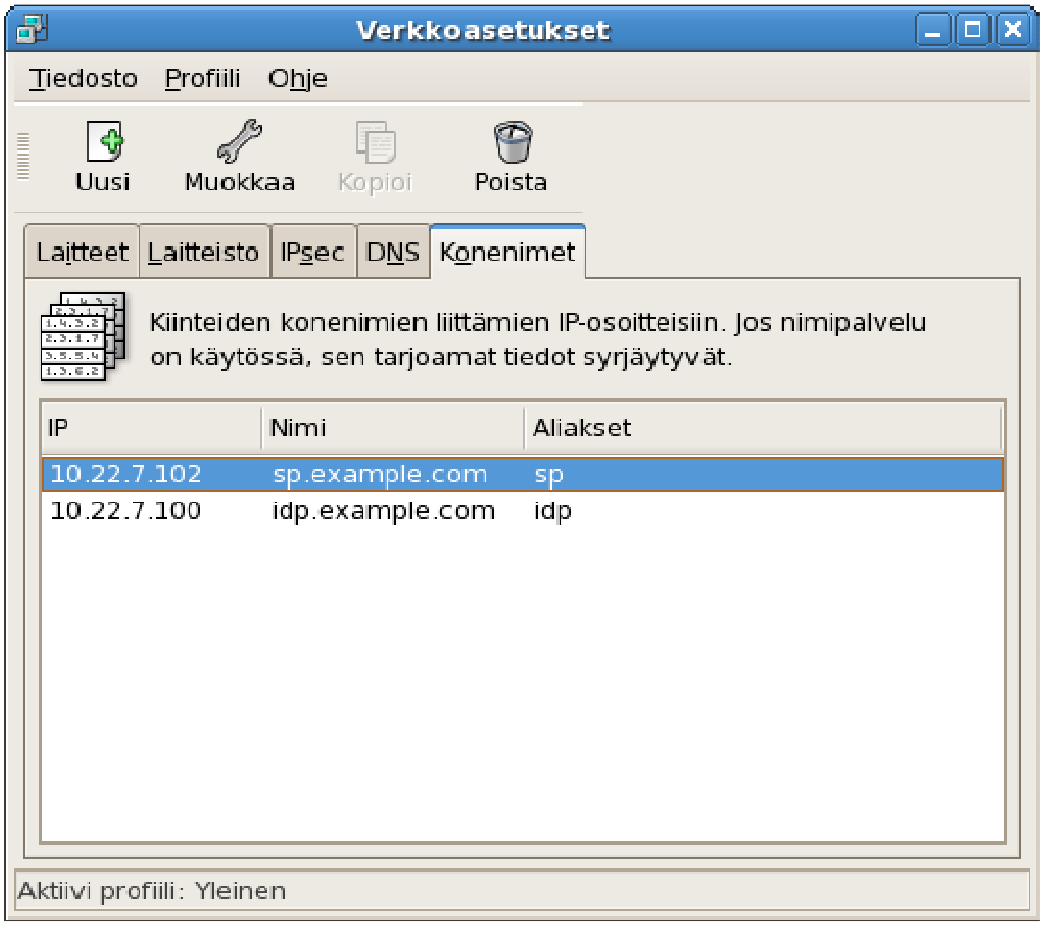
Valitaan ylävalikosta Järjestelmä → Ylläpito → Verkko

Laitteet välilehden ollessa näkyvissä valitaan Muokkaa. Yleinen- välilehdellä laitetaan täppä kohtaan Staattiset IP-osoitteet, eli määritämme tässä koneelle staattisen IP:n emmekä käytä DHCP palvelinta IP:n hankkimiseen. Asennuksessa käyttämämme IP-osoite on varattu Laurean verkosta käyttöömme. Osoiteriville kirjoitetaan IP, eli tässä tapauksessa 10.22.7.102 ja aliverkoksi 255.255.252.0. Oletusyhdyskäytävänä käytetään osoitetta 10.22.4.1. Tämän jälkeen varmistetaan valinnat painamalla OK.



Kuvio 16: Verkkoasetukset

Seuraavaksi Verkkoasetukset-ikkunasta valitaan välilehti Konenimet, johon kirjoitetaan nimiosoitteet, jotka vastaavat asennuksessa käytettäviä IP-osoitteita. Painetaan Konenimet välilehden ollessa näkyvillä Uusi. Ilmestyvään ikkunaan kirjoitetaan osoitteen kohdalle 10.22.7.102 ja konenimen kohdalle sp.example.com. Aliakseen voi halutessa kirjottaa esim. lyhenteen sp. Vahvistetaan painamalla OK ja painetaan uudestaan Uusi. Seuraavaksi ikkunaan kirjoitetaan IdP:n tiedot, eli tässä tapauksessa osoitteeksi 10.22.7.100 ja konenimeksi idp.example.com, vahvistetaan OK:lla. Suljetaan ikkuna valitsemalla Tiedosto → lopeta, ja tallennetaan tehdyt muutokset.



Kuvio 17: Verkkoasetukset (2)

8.2.2 SSL-sertifikaatti

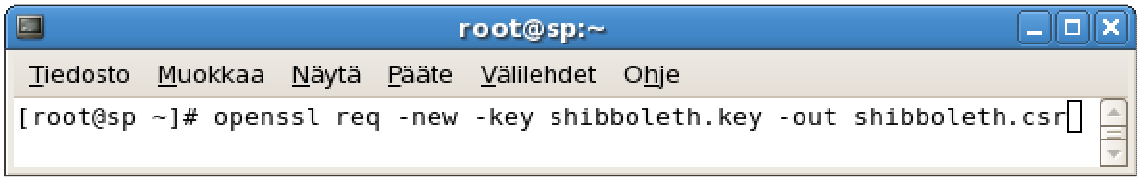
Seuraavaksi luodaan Apache WWW-palvelimella käytettävä salausavain ja sertifikaatti.

Valitaan ylävalikosta Sovellukset → Apuohjelmat → Pääte

Kirjoitetaan komentoriville seuraavat komennot:

Kuvio 18: Salausavainparin luonti

Komento luo 2048 bittisen SSL salausavainparin jonka nimi on shibboleth.key.



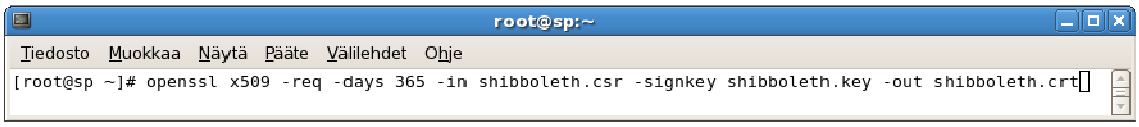
Kuvio 19: Palvelinvarmennepyynnön luominen

Komento luo luodusta avainparista palvelinvarmennepyynnön.

Ruutuun ilmestyviin kysymyksiin vastaamme tässä asennuksessa seuraavasti:

Country Name : FI
State or Providence Name : Uusimaa
Locality Name : Espoo
Organization Name : Laurea
Organizational Unit Name : Tyhjä(enter)
Common name : sp.example.com
email Address : tyhjä(enter)
A challenge Password : salasana
An optional company name : tyhjä(enter)

Seuraavaksi allekirjoitamme itse luodun palvelinvarmenteen kirjoittamalla päätteeseen seuraavan komennon:



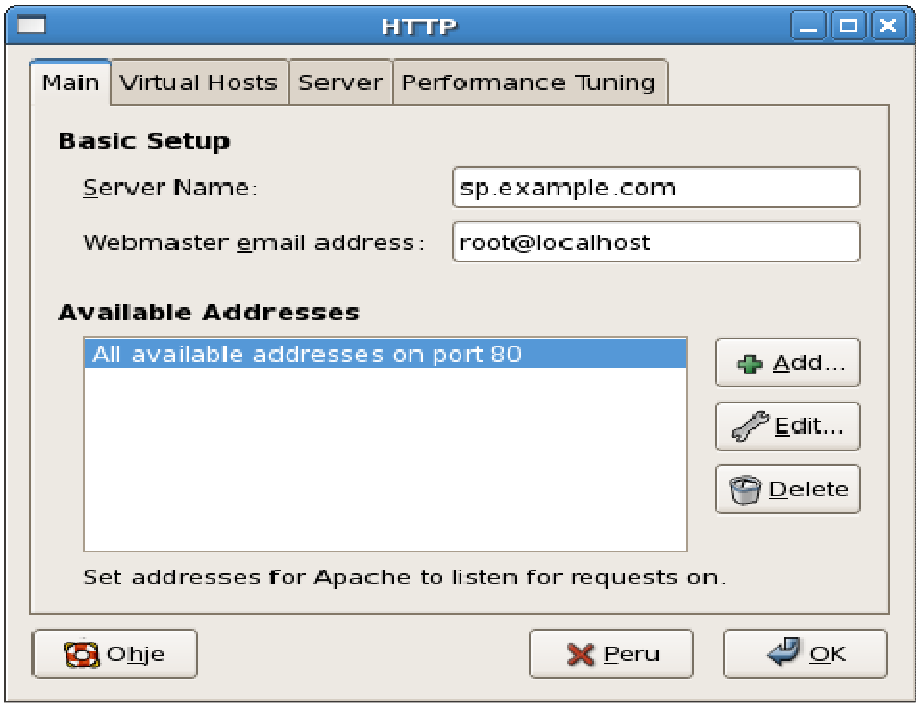
Kuvio 20: Palvelinvarmenteen allekirjoitus

Allekirjoitus on voimassa 365 päivää eli vuoden.

8.2.3 Apache

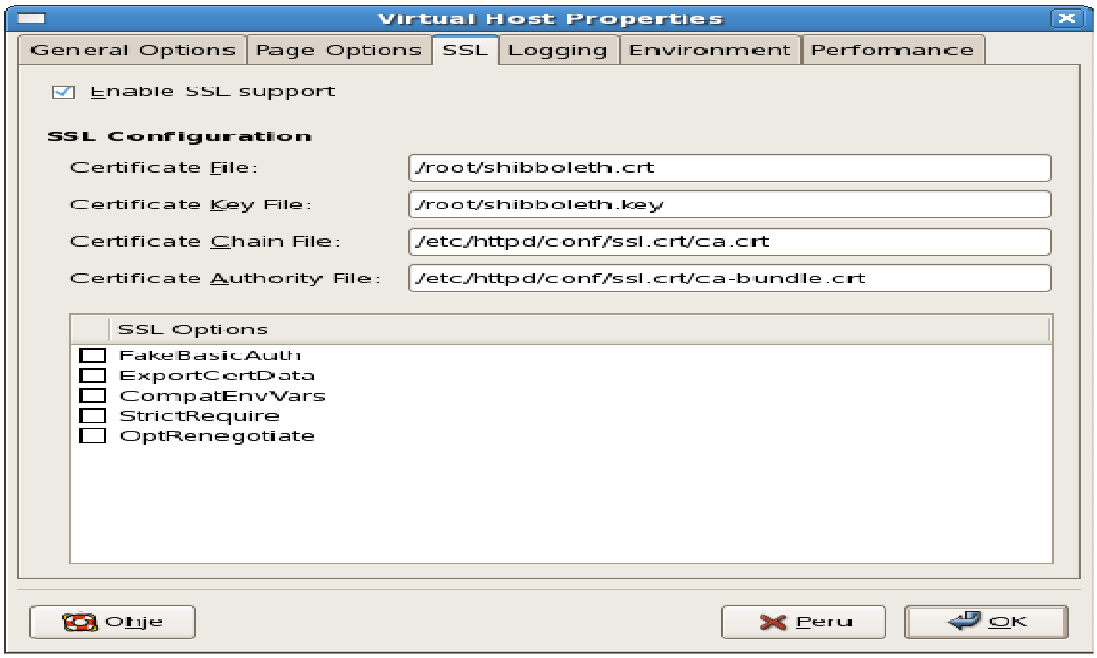
Seuraavaksi konfiguroidaan Apache WWW-palvelinta. Valitaan ylävalikosta Järjestelmä → Ylläpito → Palvelinasetukset → HTTP

Main-välilehdelle Server Name kohtaan kirjoitetaan sp.example.com.



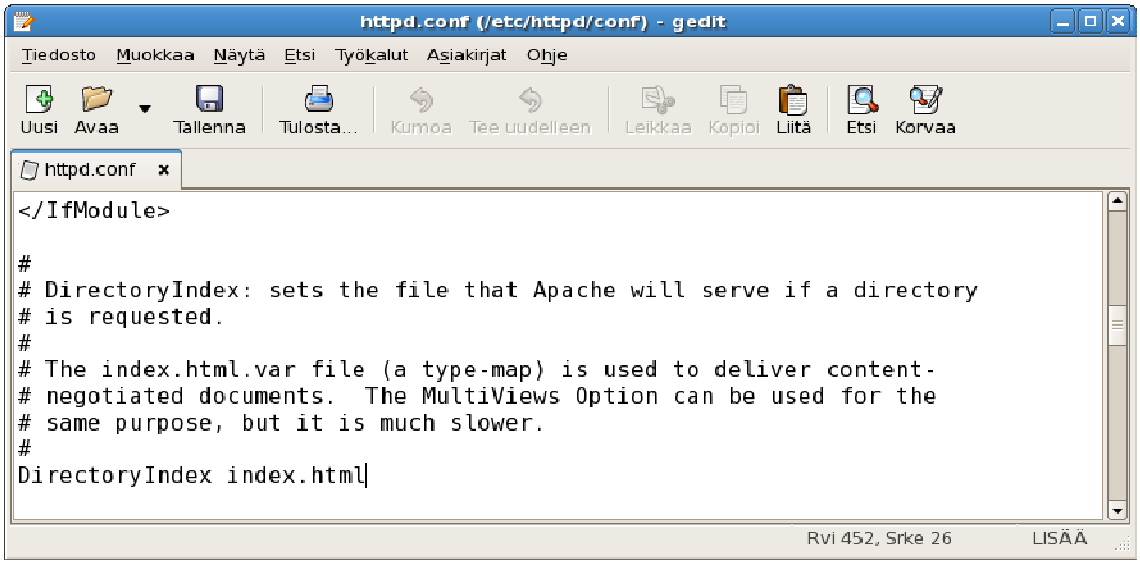
Kuvio 21: Apache-palvelimen nimeäminen

Seuraavaksi valitaan välilehti Virtual Hosts josta painetaan Edit. Valitaan välilehti SSL ja kirjoitetaan Certificate File kohtaan .crt tiedoston sijainti, /root/shibboleth.crt ja Certificate Key File kohtaan .key tiedoston sijainti, /root/shibboleth.key. Varmistetaan valinnat painamalla OK, OK ja tallennetaan valinnat.



Kuvio 22: Sertifikaattien käytäntöönpano Apachessa

Avataan tekstieditoriin httpd.conf tiedosto joka löytyy hakemistosta /etc/httpd/conf/
Lisätään ensimmäiseen DirectoryIndex kohtaan teksti index.html



Kuvio 23: Apachen pääsivu määritys

Apachen voi käynnistää valitsemalla ylävalikosta:

Järjestelmä → Ylläpito → Palvelinasetukset → Palvelut

Esiin tulevalta listalta löytyy httpd-palvelu, jonka käynnistämällä Apache käynnistyy.

8.2.4 Service Provider

Ladataan ja asennetaan Shibboleth Service Provider paketit. Mennään internet selaimella osoitteeseen:

http://shibboleth.internet2.edu/downloads/shibboleth/cppsp/2.2/RPMS/i386/RHE/5/

Sivulta ladataan kaikki .rpm paketit, joiden nimessä ei esiinny sanoja debuginfo, devel tai doc, eli yhteensä 6 asennuspakettia. Paketit ladataan ja asennetaan samalla graafisesti. Tämän asennuksen aikaiset sivulta löytyvät versiot:

log4shib-1.0.2-1.i386.rpm

opensaml-2.2-1.i386.rpm
shibboleth-2.2-3.i386.rpm
xerces-c-3.0.1-1.i386.rpm
xml-security-c-1.5.0-1.i386.rpm
xmltooling-1.2-1.i386.rpm

Shibbolethin voi käynnistää valitsemalla ylävalikosta:

Järjestelmä → Ylläpito → Palvelinasetukset → Palvelut

Esiin tulevalta listalta löytyy shibd-palvelu, jonka käynnistämällä shibboleth käynnistyy.

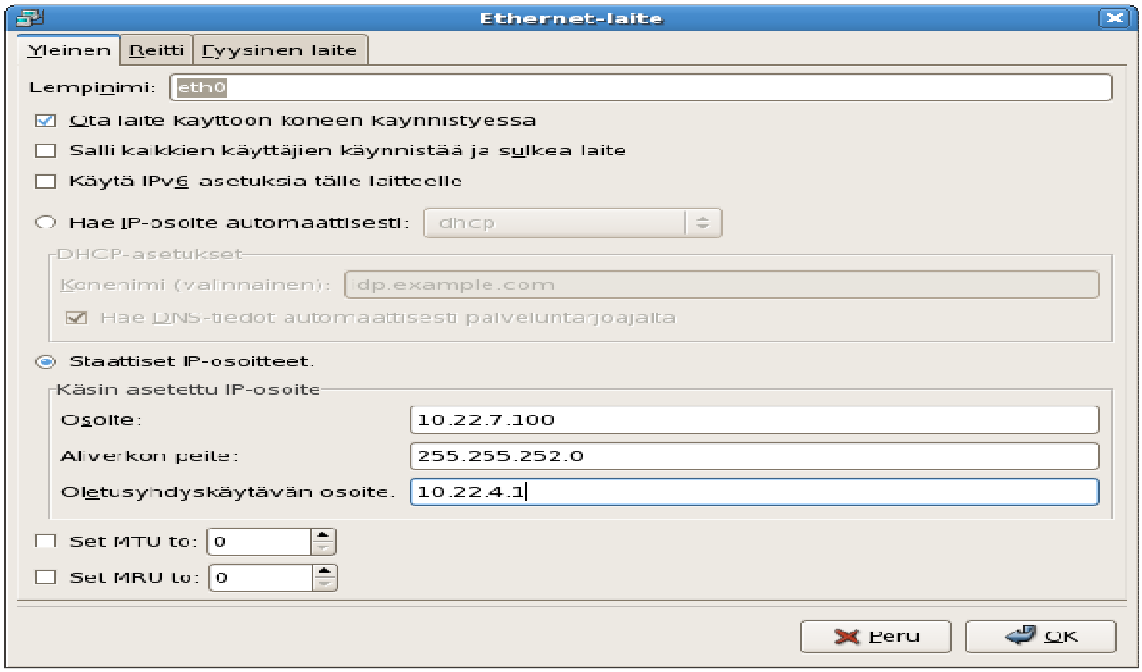
8.3 Identity Provider-asennus

Tässä osiossa asennamme toiselle serverille Identity Providerin. Ennen Shibboleth Identity Providerin asennusta tehdään muutamia esivalmisteluja.

8.3.1 Verkkoasetukset

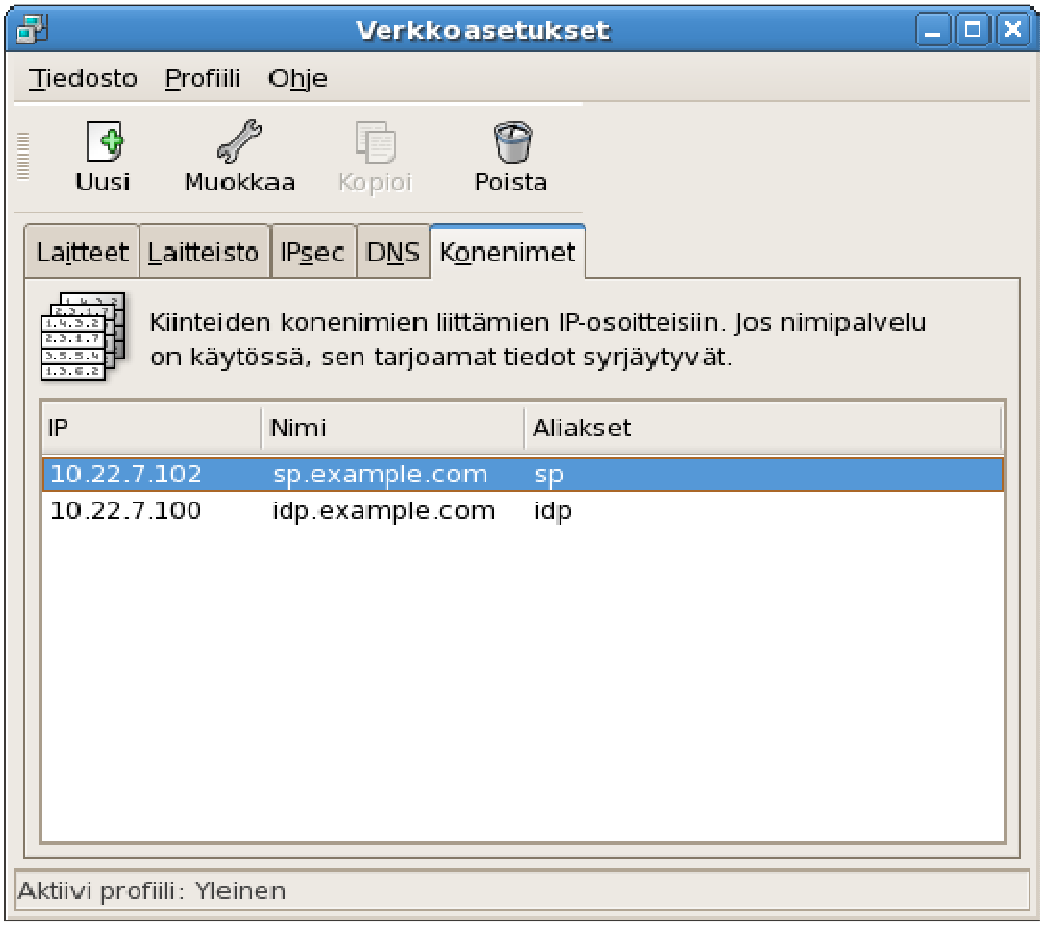
Valitaan ylävalikosta Järjestelmä → Ylläpito → Verkko

Laitteet välilehden ollessa näkyvissä valitaan Muokkaa. Yleinen- välilehdellä laitetaan täppä kohtaan Staattiset IP-osoitteet, eli määritämme tässä koneelle staattisen IP:n emmekä käytä DHCP-palvelinta IP:n hankkimiseen. Asennuksessa käyttämämme IP-osoite on varattu Laurean verkosta käyttöömme. Osoiteriville kirjoitetaan IP, eli tässä tapauksessa 10.22.7.100 ja ali-verkoksi kirjotetaan 255.255.252.0. Oletusyhdyskäytävänä käytetään osoitetta 10.22.4.1. Tämän jälkeen varmistetaan valinnat painamalla OK.



Kuvio 24: IP-osoitteiden muokkaus

Seuraavaksi Verkkoasetukset ikkunasta valitaan välilehti Konenimet, johon kirjoitetaan nimiosoitteet jotka vastaavat asennuksessa käytettäviä IP-osoitteita. Painetaan Konenimet-välilehden ollessa näkyvillä Uusi. Ilmestyvään ikkunaan kirjoitetaan osoitteen kohdalle 10.22.7.102 ja konenimen kohdalle sp.example.com. Aliakseen voi halutessa kirjottaa esim. lyhenteen sp. Vahvistetaan painamalla OK ja painetaan uudestaan Uusi. Seuraavaksi ikkunaan kirjoitetaan IdP:n tiedot, eli tässä tapauksessa osoitteeksi 10.22.7.100 ja konenimeksi idp.example.com, vahvistetaan OK:lla. Suljetaan ikkuna valitsemalla Tiedosto → lopeta ja tallennetaan tehdyt muutokset.



Kuvio 25: Verkkoasetukset

8.3.2 Java

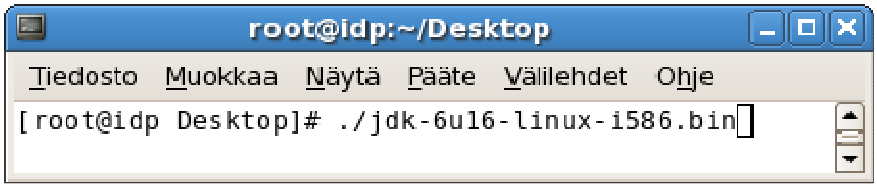
IdP vaatii toimiakseen JVM:n, eli Java Virtual Machinen. Asennetaan JVM siirtymällä internet-selaimella osoitteeseen:

http://java.sun.com/javase/downloads/index.jsp

Valitaan uusin Java SE Development Kit, tässä tapauksessa JDK 6 Update 16 ja painetaan Download. Valitaan käyttöjärjestelmä mihin aseenus suoritetaan, eli tässä tapauksessa Linux ja hyväksytään käyttösäännöt. Klikataan linkin kohdalla jossa ei lue rpm ja ladataan tiedosto.

Siirrytään päätteeseen ja mennään hakemistoon jossa ladattu tiedosto sijaitsee. Kirjoitetaan komentoriville seuraavat komennot:

Kuvio 26: käyttöoikeuksien anto tiedostolle



Kuvio 27: Asennuspaketin suorittaminen

Hyväksytään käyttösäännöt kirjoittamalla yes ja painetaan enter.

Määritetään Javan ympäristömuuttuja:

Kuvio 28: Java ympäristömuuttujan määritys

8.3.3 Apache Tomcat

Asennetaan Apache Tomcat siirtymällä internet selaimella osoitteeseen:

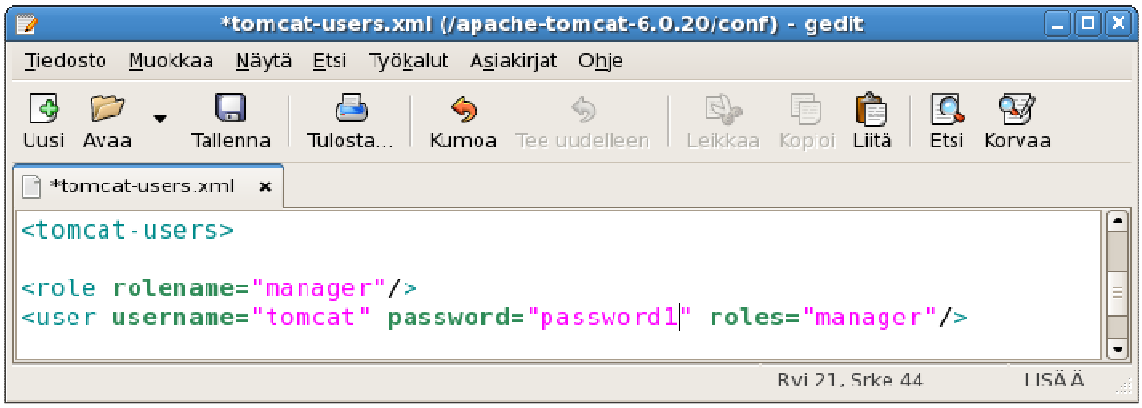
http://tomcat.apache.org

Valitaan vasemmalta uusin versio, tässä tapauksessa 6.x. Klikataan Binary Distributions → Core kohdassa zip linkkiä ja ladataan ja puretaan paketti.

Annetaan Tomcatille käytettäväksi riittävä määrä muistia kirjoittamalla päätteeseen seuraava komento:

Kuvio 29: Tomcat:in muistin lisääminen

Tehdään Tomcatille manager käyttäjä, jolla voi hallita sovelluksia graafisessa käyttöliitymässä. Avataan tekstieditoriin tomcat-users.xml joka sijaitsesee Tomcatin conf kansiossa. Tehdään siihen seuraavanlaiset muutokset:

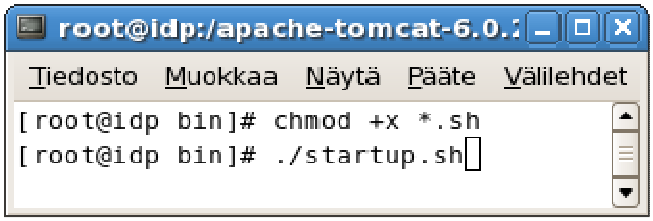


Kuvio 30: Käyttäjien luonti

Annetaan Javalle ympäristömuuttuja jotta Tomcat tunnistaa sen:

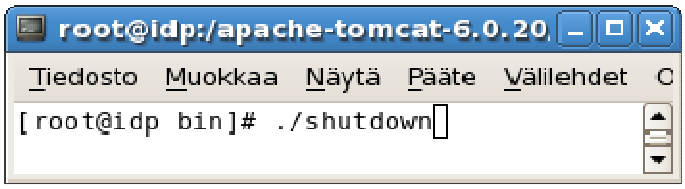
Kuvio 31: Java ympäristömuuttujan asetus

Tomcatin voi käynnistää päätteestä menemällä Tomcatin kansioon bin ja suorittamalla seuraavat käskyt:



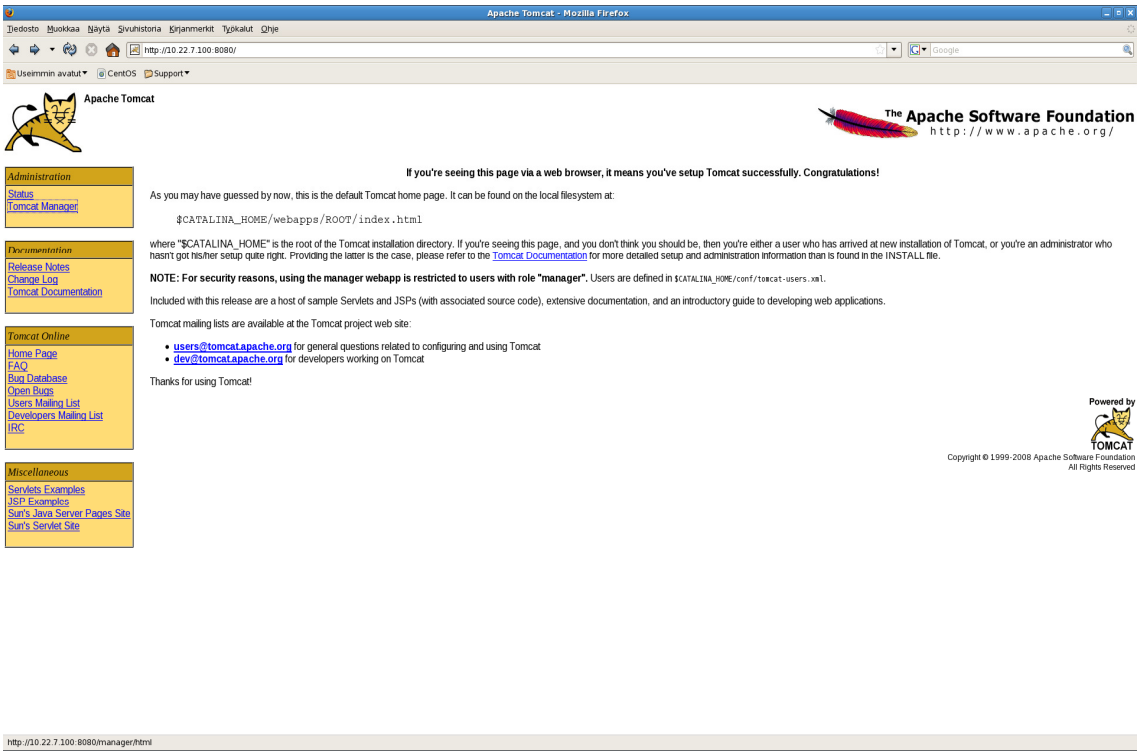
Kuvio 32: Tiedoston käyttöoikeuksien anto sekä Tomcat:in käynnistys

Tomcat sammutetaan kirjoittamalla päätteessä Tomcatin bin kansiossa:



Kuvio 33: Tomcat:in sammutus

Graafisen käyttöliittymän voi avata siirtymällä internet selaimella osoitteeseen 10.22.7.100:8080



Kuvio 34: Tomcat:in graafinen käyttöliittymä

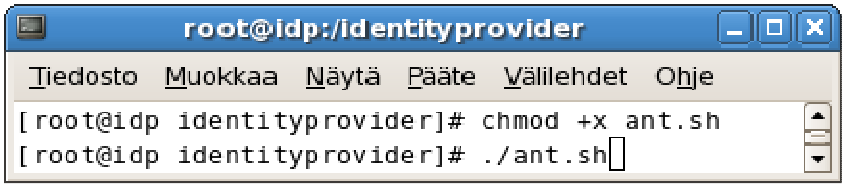
8.3.4 Identity Provider

Ladataan ja asennetaan Shibboleth Service Provider paketit. Siirrytään internet selaimella osoitteeseen:

http://shibboleth.internet2.edu/downloads/shibboleth/idp/2.0/

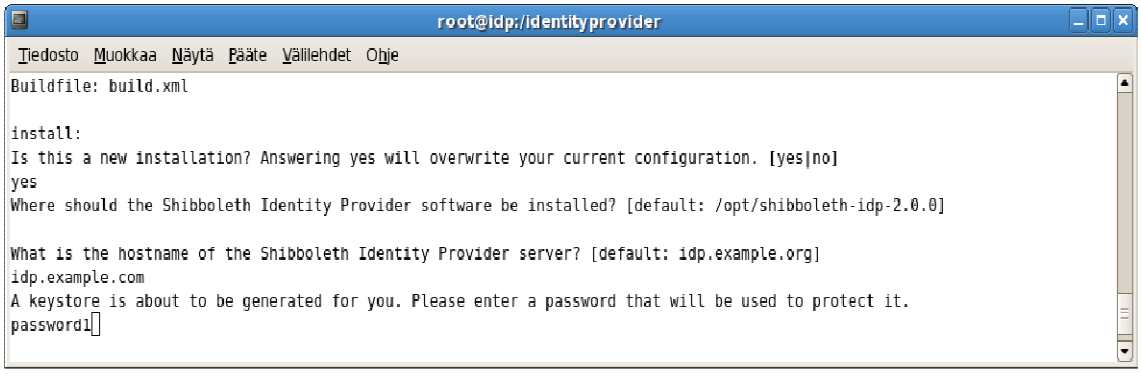
Klikataan shibboleth-idp-2.0.0-bin.zip linkkiä ja ladataan ja puretaan paketti.

Siirrytään päätteellä kansioon johon paketti purettiin ja kirjoitetaan seuraavat komennot:



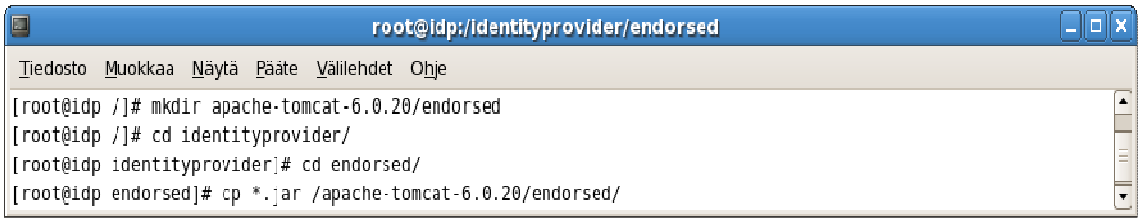
Kuvio 35: Oikeuksien anto ja asennuspaketin suorittaminen

Vastataan kysymyksiin seuraavasti:



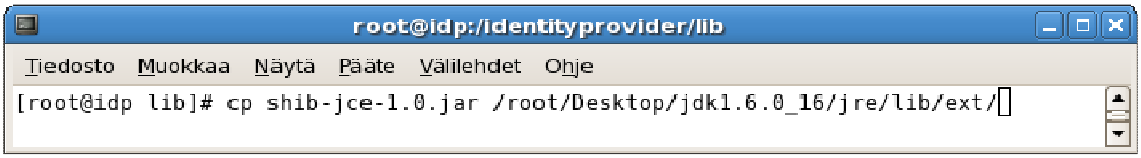
Kuvio 36: Shibboleth asennus

Asennuksen jälkeen luodaan Tomcatin kansioon hakemisto endorsed ja kopioidaan sinne IdP:n asennuspakettikansion hakemistosta endorsed löytyvät .jar tiedostot:



Kuvio 37: Tiedostojen kopiointi

Kopioidaan IdP:n asennuspakettikansion lib hakemistosta löytyvä shib-jce-1.0.jar Javan kansion hakemistoon jre/lib/ext.

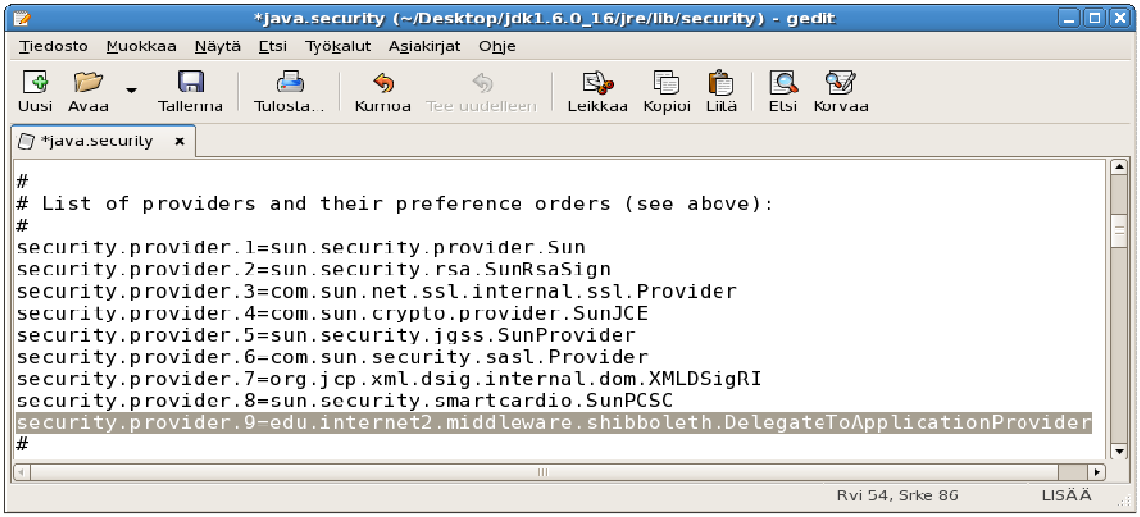


Kuvio 38: Jar paketin kopiointi

Avataan nano tekstieditoriin java.security tiedosto joka löytyy Javan kansiosta jre/lib/security.

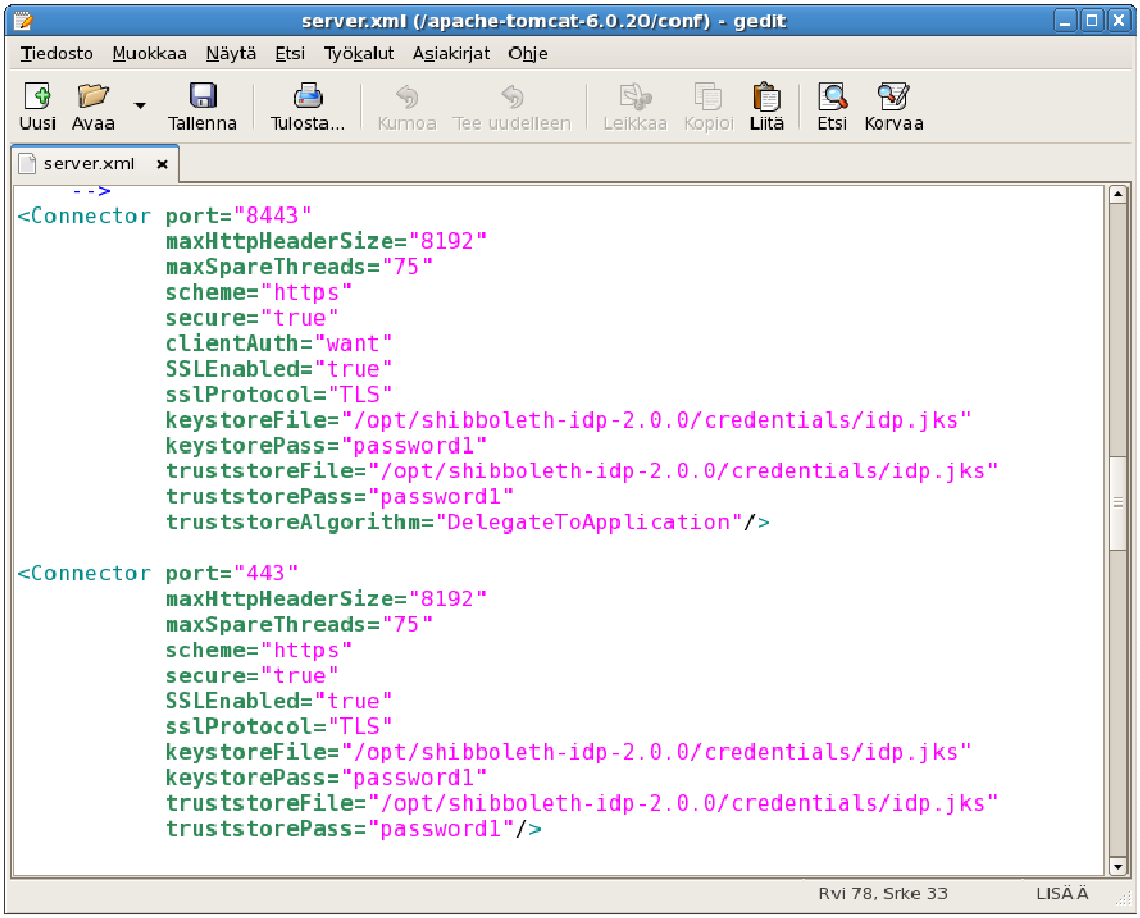
Kuvio 39: java.securityn muokkaus

Lisätään siihen seuraava rivi ja tallennetaan muutokset:



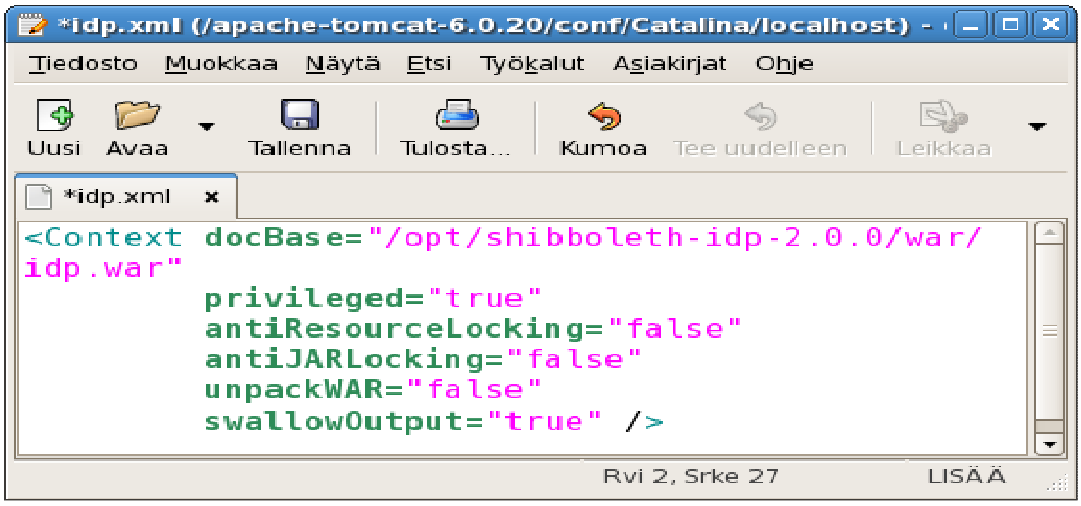
Kuvio 40: Java.securityn muokkaus (2)

Avataan graafiseen tekstieditoriin Tomcatin conf hakemistosta löytyvä server.xml ja lisätään sinne seuraavat connectorit, lisäykset pitää tehdä joko ensimmäiseksi tai viimeiseksi connector listaan :



Kuvio 41: Server.xml

Luodaan Tomcatin kansioon conf/Catalina/localhost kansioon tiedosto idp.xml, kirjoitetaan siihen seuraavat rivit ja tallennetaan:

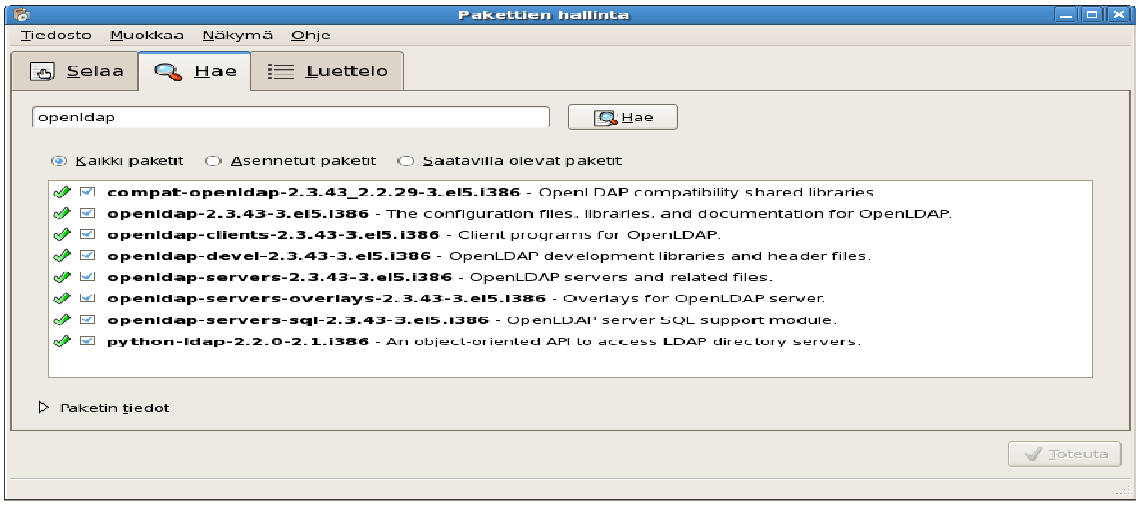


Kuvio 42: Idp.xml

8.4 LDAP-asennus

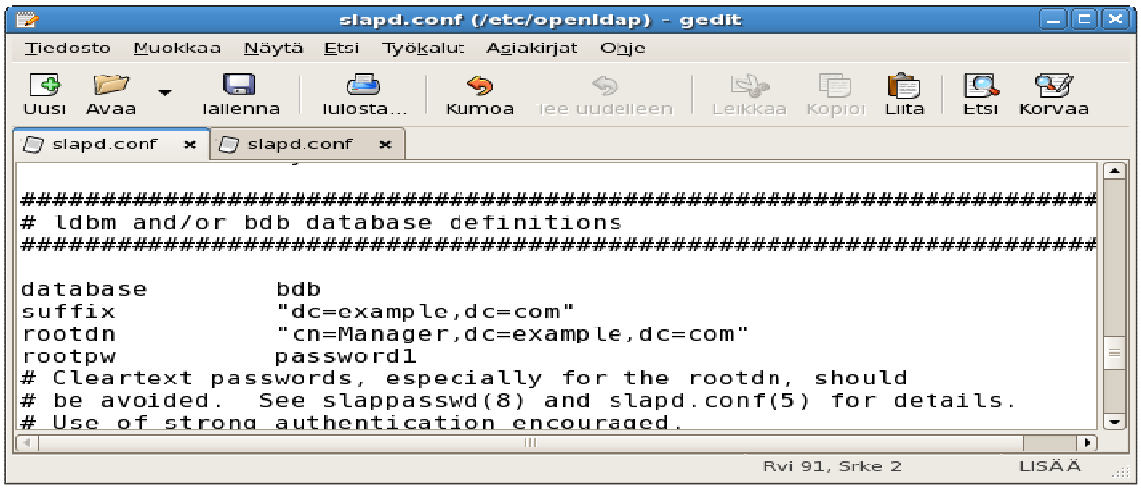
Tässä asennuksessa asennamme LDAP-serverin samalle koneelle kuin Service Providerin.

LDAP-palvelimen ohjelmistoksi valitsemme openldap 2.3:sen. Ladataan ja asennetaan asennuspaketit avaamalla ylävalikosta Sovellukset → lisää tai poista ohjelmistoja. Valitaan Hae välilehti ja kirjoitetaan hakusanaksi openldap. Merkitään kuvan osoittamat paketit ja painetaan toteuta.



Kuvio 43: Pakettien hallinta

Avataan tekstieditoriin slapd.conf tiedosto joka sijaitsee kansiossa etc/openldap ja tehdään siihen seuraavat muutokset:



Kuvio 44: Slapd.conf

Suffix määrittää domainin, tässä tapauksessa example.com, rootdn määrittää pääkäyttäjän domainiin ja rootpw pääkäyttäjän salasanan.

Kopioidaan /etc/openldap kansiossa sijaitseva DB_CONFIG.example tiedosto /var/lib/ldap kansioon ja nimetään se muotoon DB_CONFIG.

Kuvio 45: DB_CONFIG.example kopiointi

Käynnistetään openldap valitsemalla ylävalikosta:

Järjestelmä → Ylläpito → Palvelinasetukset → Palvelut

Valitaan listasta ldap ja painetaan käynnistä.

Luodaan tiedostojärjestelmän juureen / tiedosto base.ldif ja avataan se tekstieditoriin. Kirjoitetaan tiedostoon seuraavat rivit ja tallennetaan se:

Kuvio 46: Base.ldif

Tuodaan base.ldif ldap hakemistoon. Avataan pääte ja kirjotetaan siihen seuraava käsky:

Kuvio 47: Käyttäjän luonti LDAP tietokantaan

8.5 Jxplorer-asennus

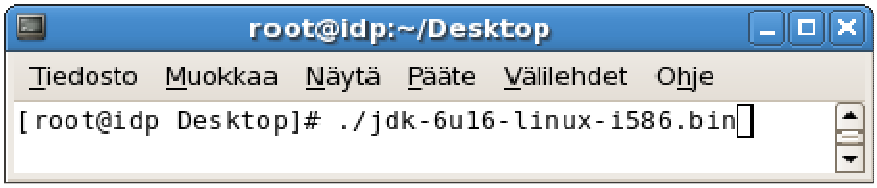
Jxplorer vaatii toimiakseen Javan. Asennetaan Java siirtymällä internet selaimella osoitteeseen:

http://java.sun.com/javase/downloads/index.jsp

Valitaan uusin Java SE Development Kit, tässä tapauksessa JDK 6 Update 16 ja painetaan Download. Valitaan käyttöjärjestelmä mihin aseenus suoritetaan, eli tässä tapauksessa linux ja hyväksytään käyttösäännöt. Klikataan linkin kohdalla jossa ei lue rpm ja ladataan tiedosto.

Siirrytään päätteeseen, ja mennään hakemistoon jossa ladattu tiedosto sijaitsee. Kirjoitetaan komentoriville seuraavat komennot:

Kuvio 48: Käyttöoikeuksien anto tiedostolle



Kuvio 49: Jxplorer asennuspaketin suorittaminen

Hyväksytään käyttösäännöt kirjoittamalla yes ja painetaan enter.

Määritetään Javan ympäristömuuttuja:

Kuvio 50: Java-ympäristömuuttujan määritys

Ladataan Jxplorer asennuspaketti siirtymällä internet selaimella osoitteeseen:

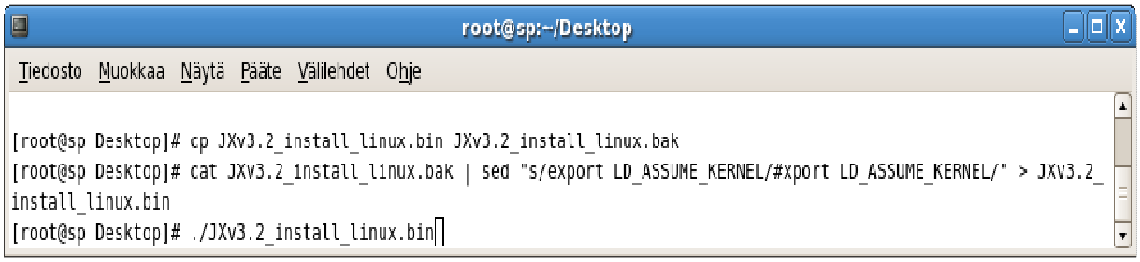
http://www.jxplorer.org

Valitaan vasemmalta:

Download → Install package

Klikataan Linuxin kohdalla download linkkiä ja ladataan tiedosto.

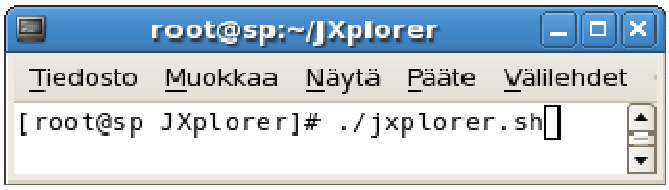
Asennetaan Jxplorer avaamalla pääte ja menemällä kansioon johon asennuspaketti on ladattu. Kirjoitetaan seuraavat komennot:



Kuvio 51: JXplorer asennuspaketin muokkaus

Asennuksen käynnistyessä pidetään oletusarvot eli painetaan nextiä kunnes asennus on valmis.

Käynnistetään Jxplorer avaamalla pääte, menemällä Jxplorerin kansioon ja kirjoittamalla seuraava komento:

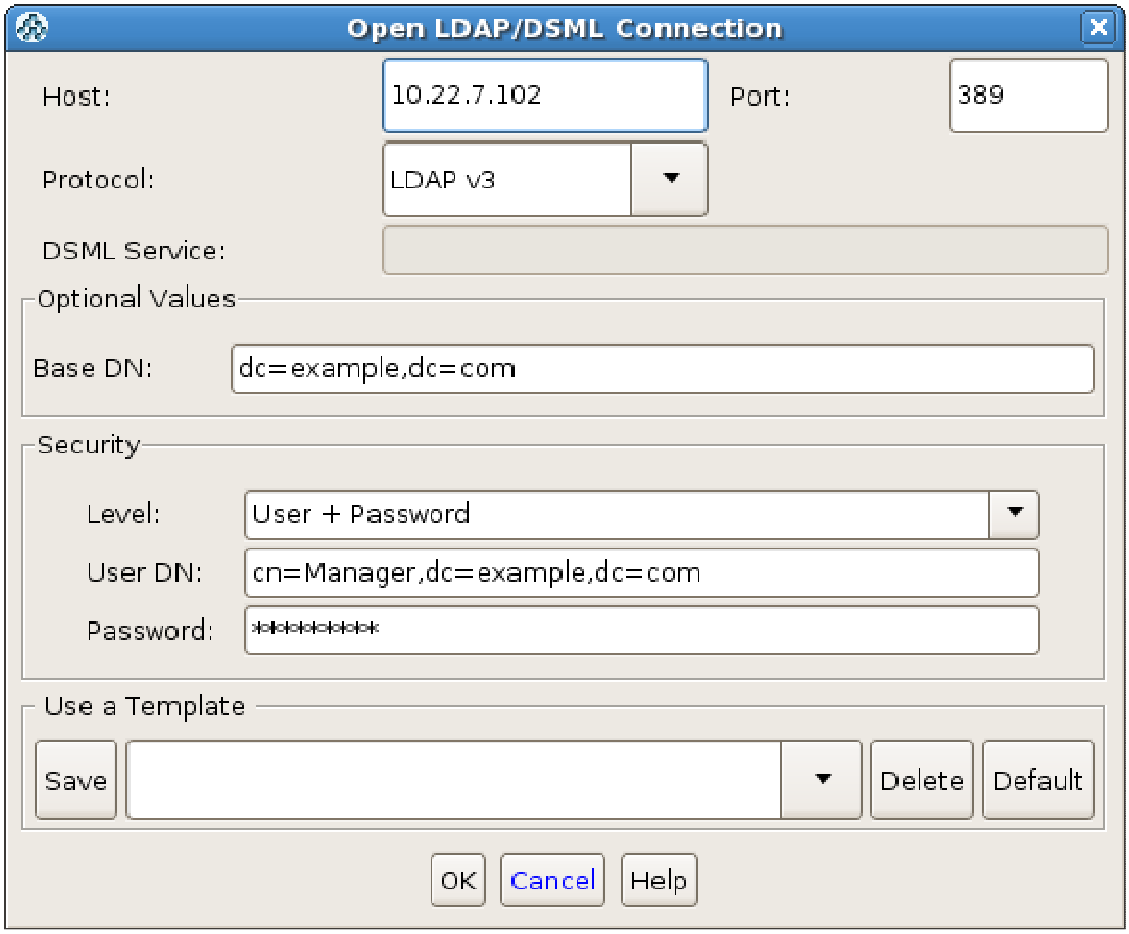


Kuvio 52: JXplorerin käynnistys

Yhdistetään ldappiin valitsemalla:

File → Connect

Tehdään ruutuun seuraavat valinnat:



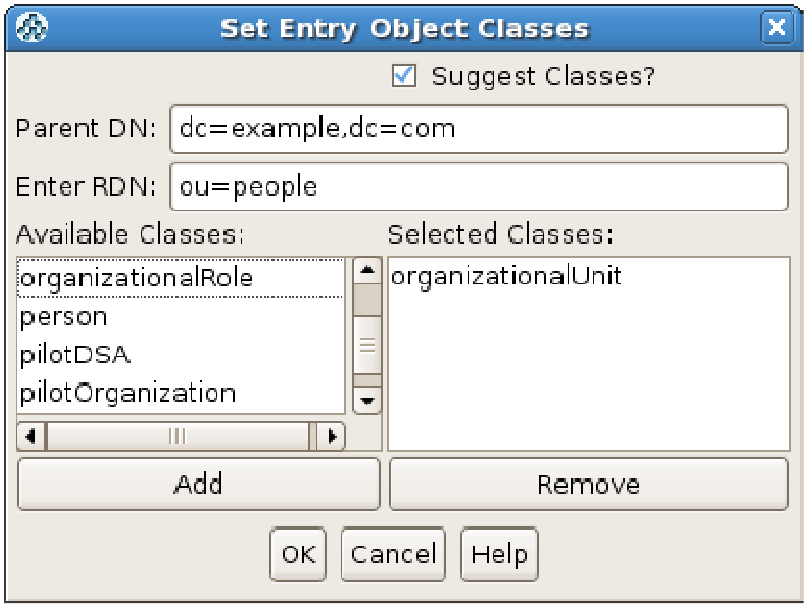
Kuvio 53: Yhteyden muodostaminen JXplorerissa

Password on LDAP:n slapd.conf tiedostoon määritetty rootpw.

Luodaan examplen alle uusi organizational unit. Valitaan ylävalikosta:

Edit → New

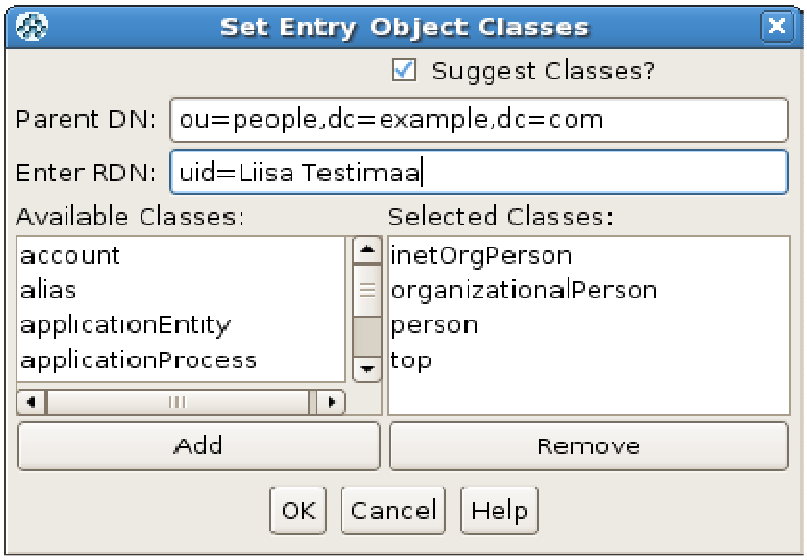
Luodaan people niminen organizationalUnit tekemällä seuraavat valinnat:



Kuvio 54: Ou:n luonti

Muokataan tietoja halutessa ja painetaan Submit.

Luodaan people organizationalUnitiin muutama käyttäjä samalla tavalla kun teimme people ou:n, vasemmalta täytyy olla valittuna people kun operaatiot tehdään:

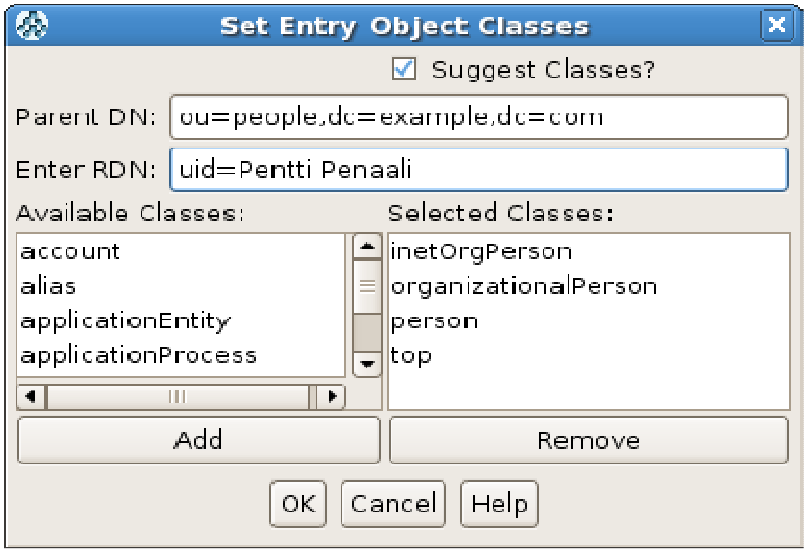


Kuvio 55: Uid:n luonti

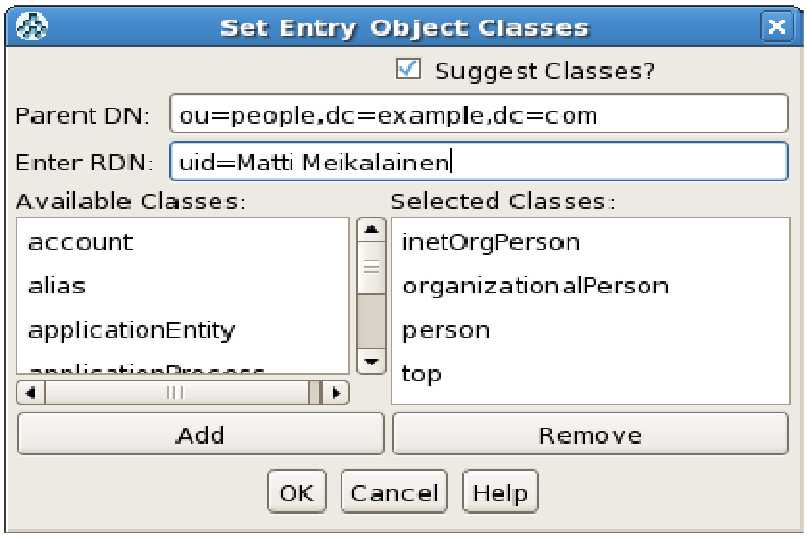
Kaavakkeesta täytyy täyttää ainakin pakolliset kohdat, eli cn(CommonName) ja sn(Surname). Täytetään lisäksi UserPassword attribuutti eli käyttäjän salasana, jonka Shibbolethin järjes-

telmä tarvitsee käyttäjän tunnistukseen. Lisäksi kaavakkeesta voi täyttää haluamiaan attribuutteja. Painetaan haluttujen muutoksien jälkeen submit.

Muut esimerkkihenkilöt:



Kuvio 56: Uid:n luonti (2)

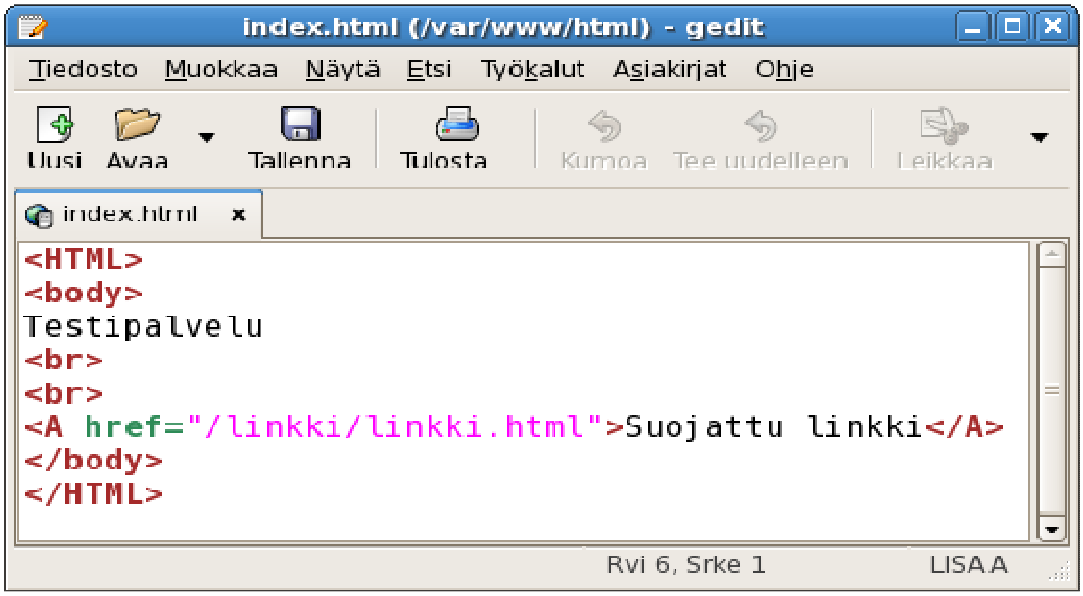


Kuvio 57: Uid:n luonti (3)

8.6 Palvelun luonti

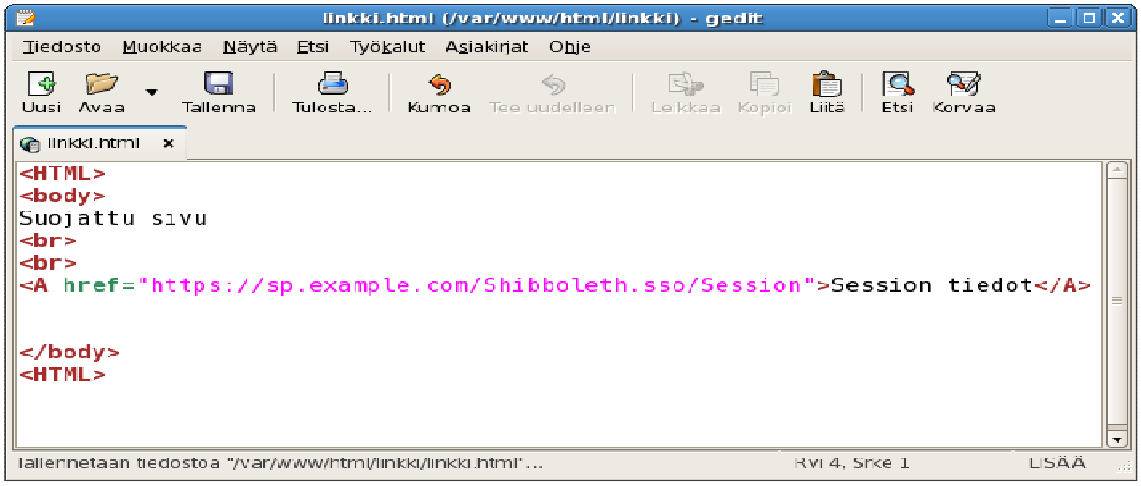
Tässä osiossa teemme ns. palvelun eli nettisivun jonka suojaamme Shibbolethilla.

Palvelu luodaan samalle koneelle johon Service Provider on asennettu.
Luodaan Apachen oletuskansioon /var/www/html tiedosto index.html. Avataan se tekstieditoriin ja muokataan se seuraavanlaiseksi:



Kuvio 58: Index.html luonti

Seuraavaksi luodaan /var/www/html kansioon uusi kansio nimeltään linkki. Luodaan linkki kansioon tiedosto linkki.html ja avataan se tekstieditoriin. Muokataan se seuraavanlaiseksi:



Kuvio 59: Linkki.html

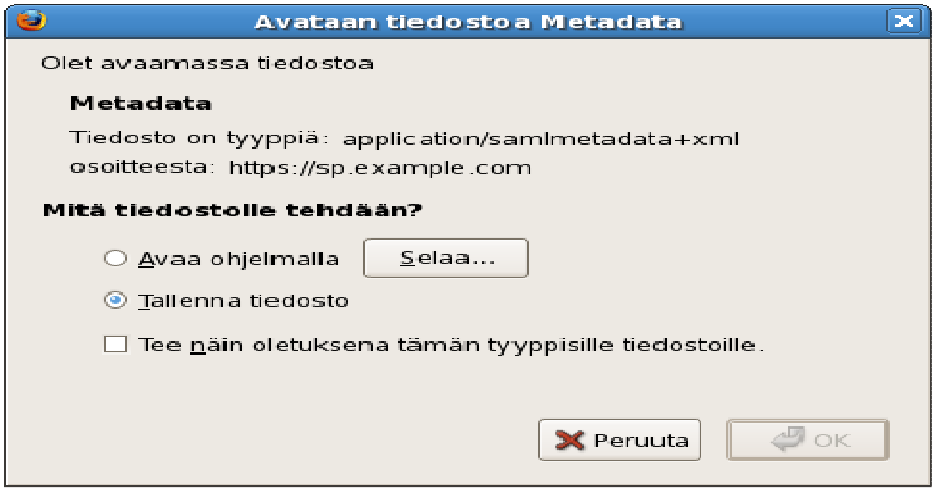
8.7 Metadatan luonti

Tässä osiossa luomme metadatan, jota Shibboleth käyttää tunnistaakseen Service Providerit ja Identity Providerit jotka keskustelevat keskenään. Tarvitsemme metadataan sekä Service Providerin että Identity Providerin tiedot. Service Providerin tiedot saamme koneelta johon se on asennettu ja Identity Providerin tiedot muokkaamalla kummaltakin koneelta löytyvän example-metadata.xml tiedoston sisältöä.

Siirrytään Service Provider koneella internet selaimella osoitteeseen:

https://sp.example.com/Shibboleth.sso/Metadata

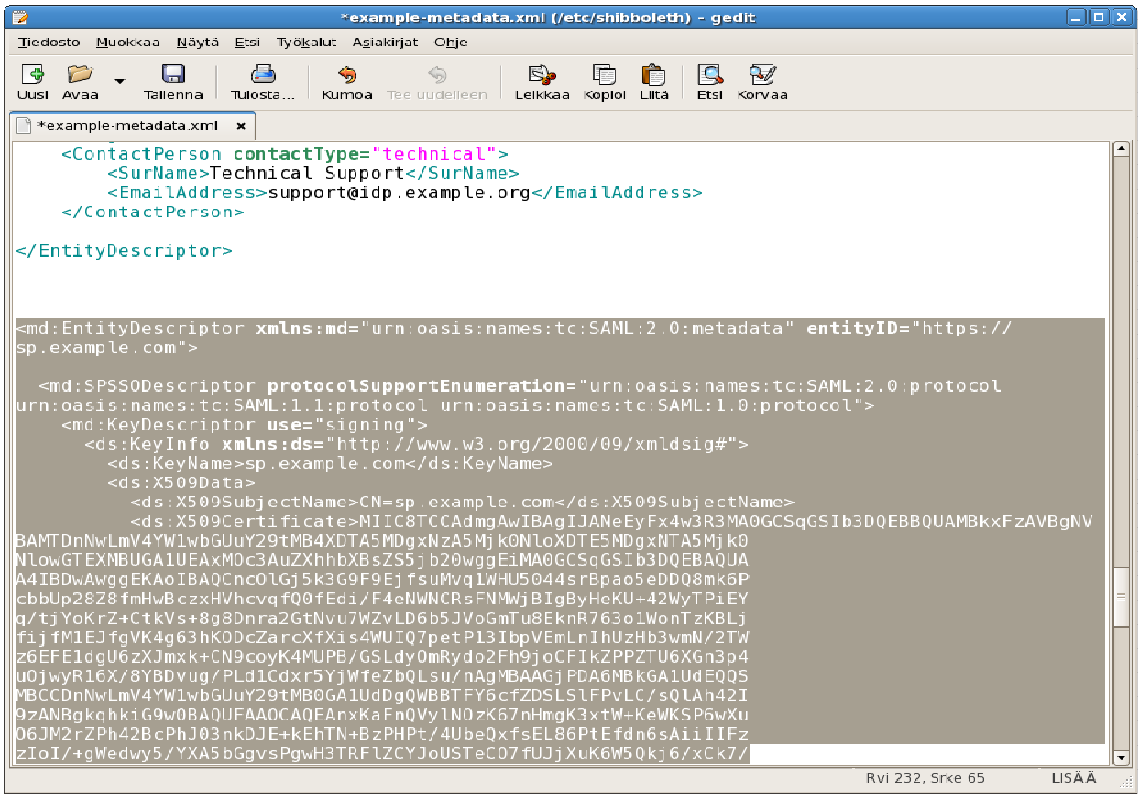
Shibd ja httpd palveluiden pitää olla päällä jotta oikea sivu tulee esiin. Tallennetaan metadata tiedosto koneelle valitsemalla Tallenna tiedosto ja OK.



Kuvio 60: Metadatan tallennus

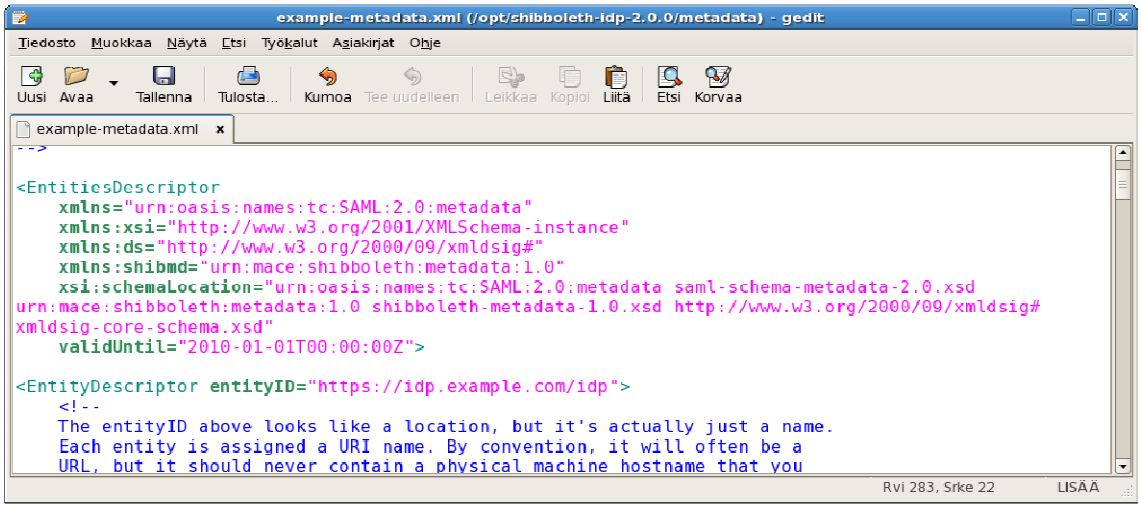
Avataan tekstieditoriin juuri ladattu tiedosto. Avataan samaan editori-ikkunaan myös example-metadata.xml tiedosto joka sijaitsee hakemistossa /etc/shibboleth.

Kopioidaan koko Metadata tiedoston sisältö ja liitetään se example-metadata.xml tiedoston loppuun, eli </Entitydescriptor> koodin jälkeen:



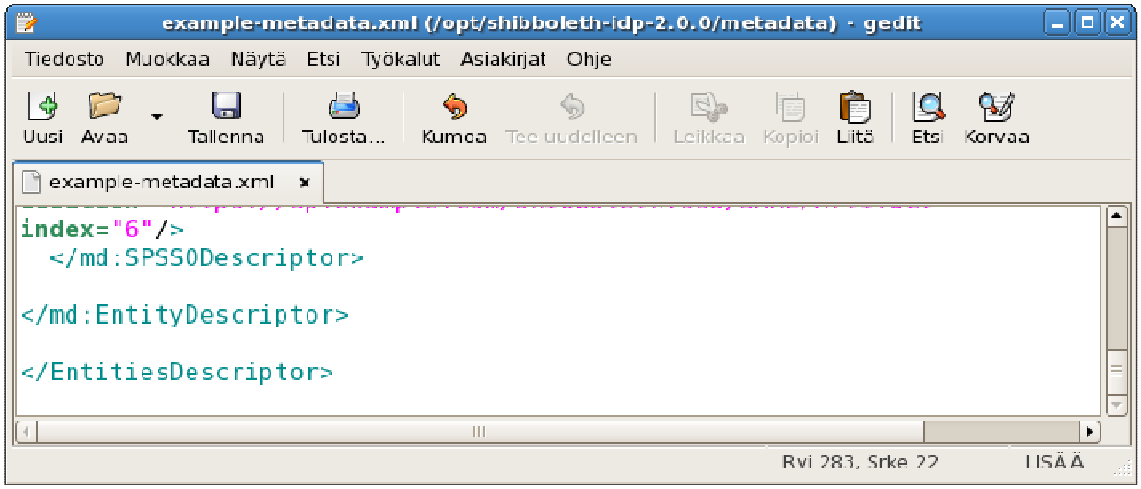
Kuvio 61: Metadatan sisällön kopiointi

Laitetaan kaikki metadatassa listatut Providerit käyttämään vaadittuja nimiavaruuksia tekemällä seuraavat muutokset. Vaihdetaan ensimmäisen EntityDescriptorin tilalle tekti Entitiesdescriptor ja lisätään > merkki validUntil määrityksen jälkeen. Lisätään teksti <EntityDescriptor ennen entityID kohtaa.



Kuvio 62: Metadatan kopiointi (2)

Lisätään koko tiedoston loppuun teksti </EntitiesDescriptor>



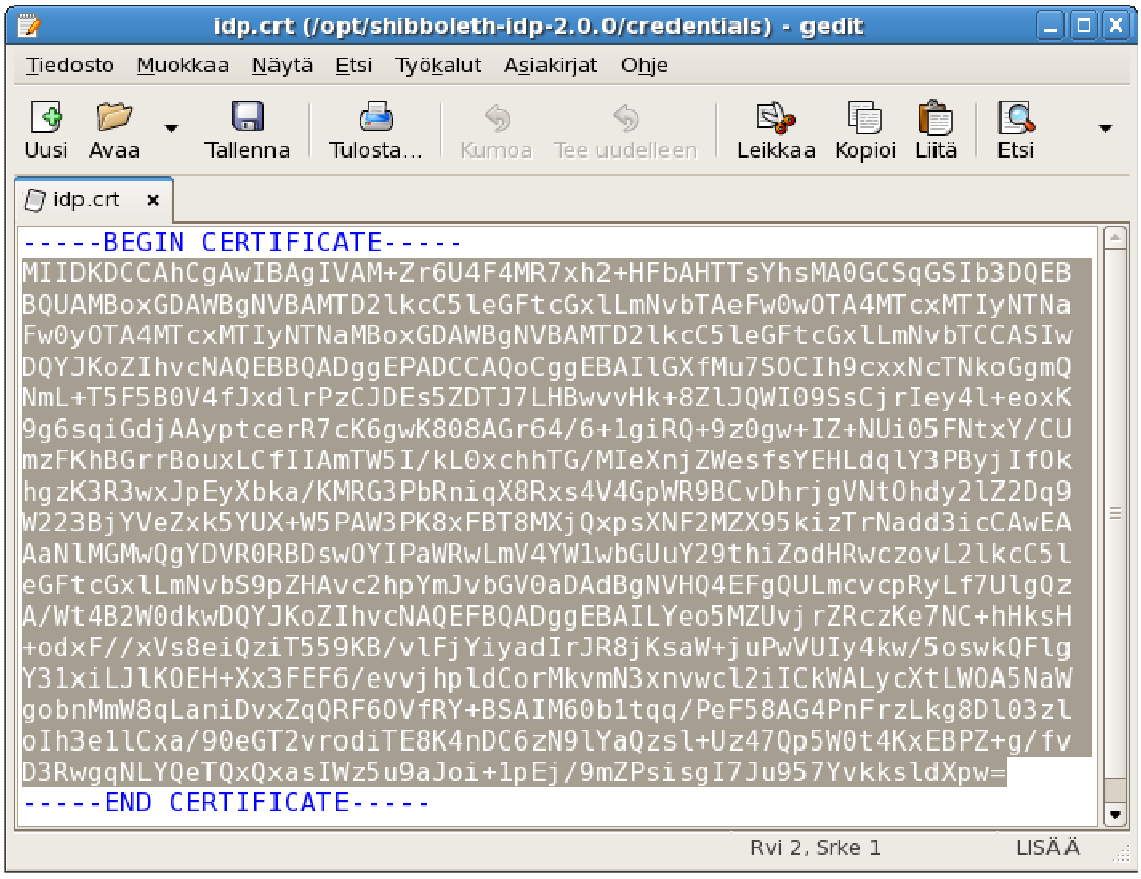
Kuvio 63: </EntitiesDescriptor>:rin lisääminen

Seuraavaksi muokkaamme example-metadata.xml tiedostoa ensimmäisen Entitydescriptorin, eli IdP:n kohdalta saadaksemme sen tiedot oikeiksi.

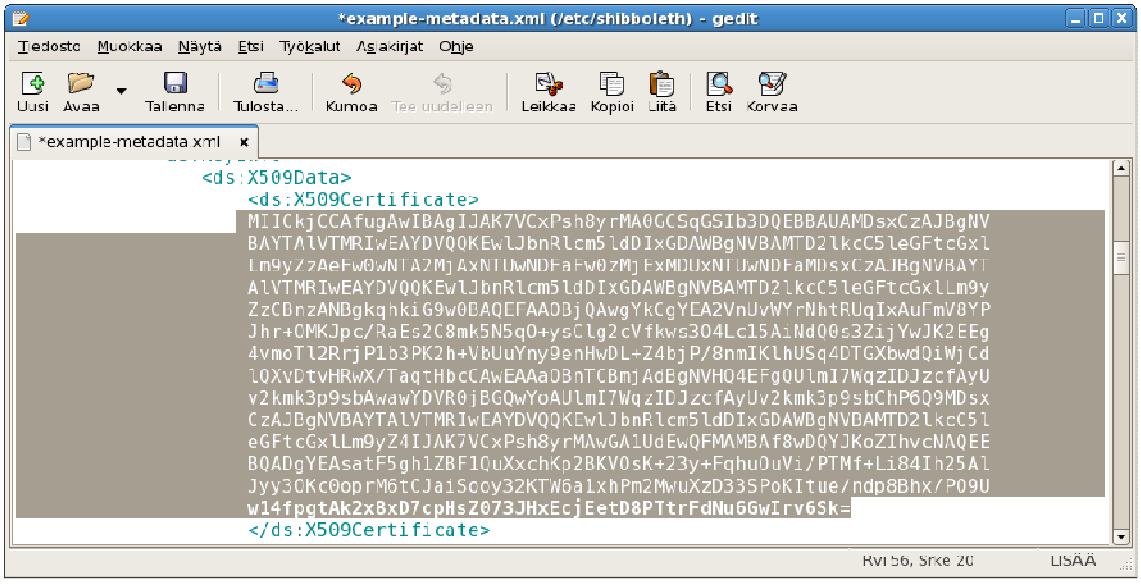
kopioidaan Identity Providerin koneelta kansiosta /opt/shibboleth-idp-2.0.0/credentials/ löytyvä idp.crt tiedosto esim. muistitikulle, jotta se saadaan siirrettyä Service Providerin koneelle.

Kopioidaan idp.crt tiedosto muistitikulta Service Provider koneen tiedostojärjestelmän juureen /.

Avataan Service Provider koneella idp.crt sekä example-metadata.xml tiedostot tekstieditoriin. Kopioidaan idp.crt tiedostosta BEGIN CERTIFICATE ja END CERTIFICATE kohtien välissä oleva teksti ja liitetään se example-metadata.xml:ssä jokaiseen ensimmäisen Entitydescriptorin x509Certificate kohtaan. Kohtia on yhteensä neljä, jokaisen kohdan koodipätkä siis korvataan idp.crt tiedoston sisältämällä koodipätkällä.

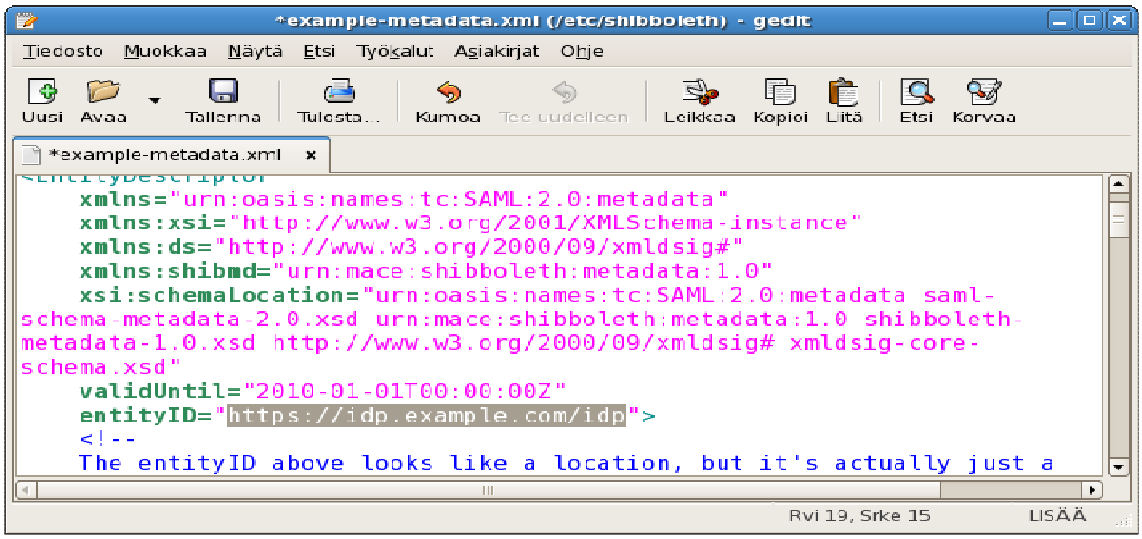


Kuvio 64: Sertifikaation kopiointi (1)

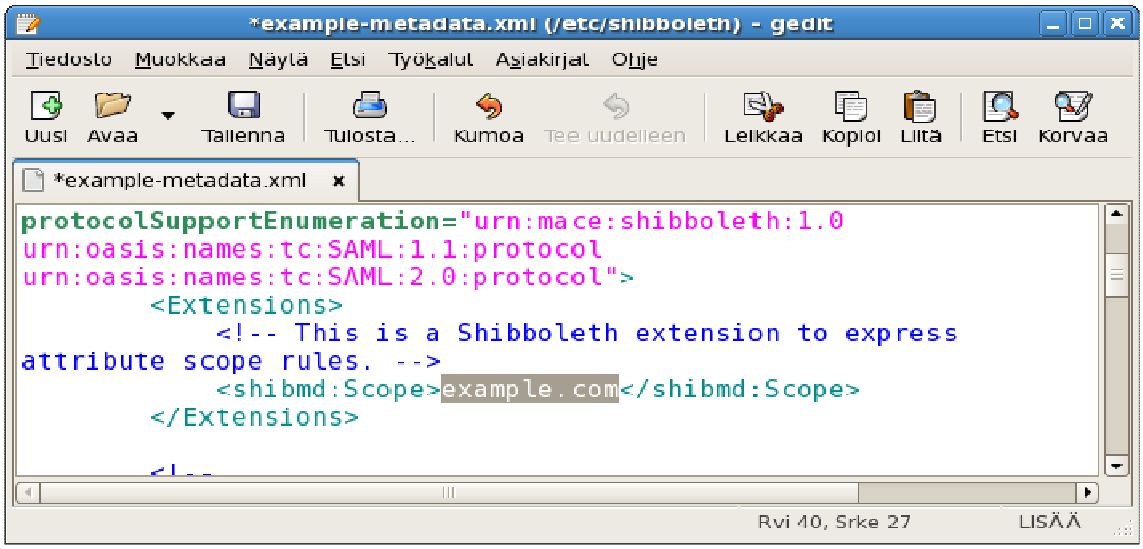


Kuvio 65: Sertifitkaatin kopiointi (2)

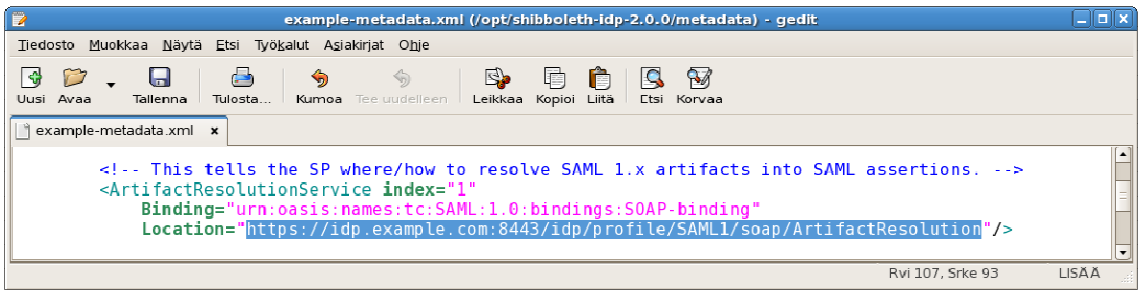
Seuraavaksi muokataan IdP:n eli ensimmäisen Entitydescriptorin muutamat kohdat vastaamaan asennuksen vaatimuksia:



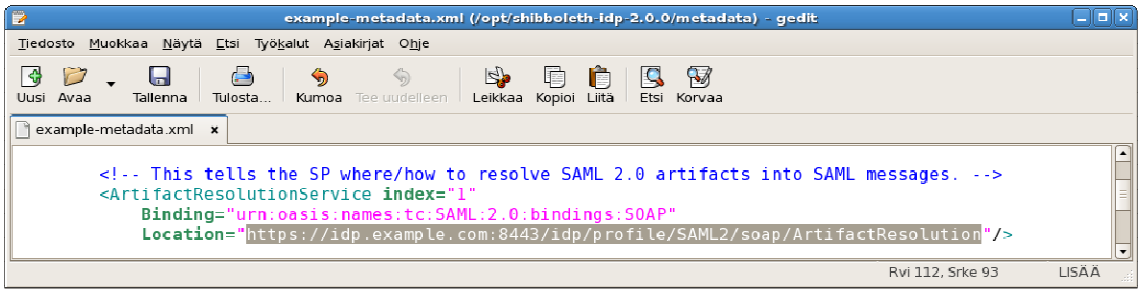
Kuvio 66: Metadatan muokkaus (1)



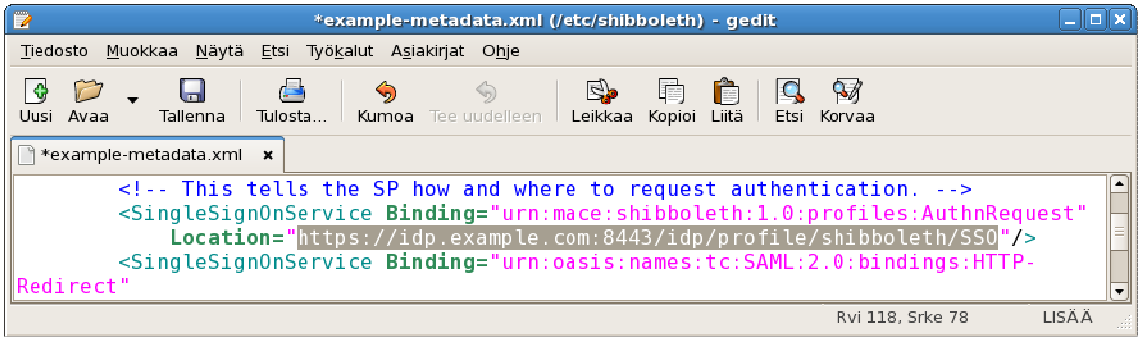
Kuvio 67: Metadatan muokkaus (2)



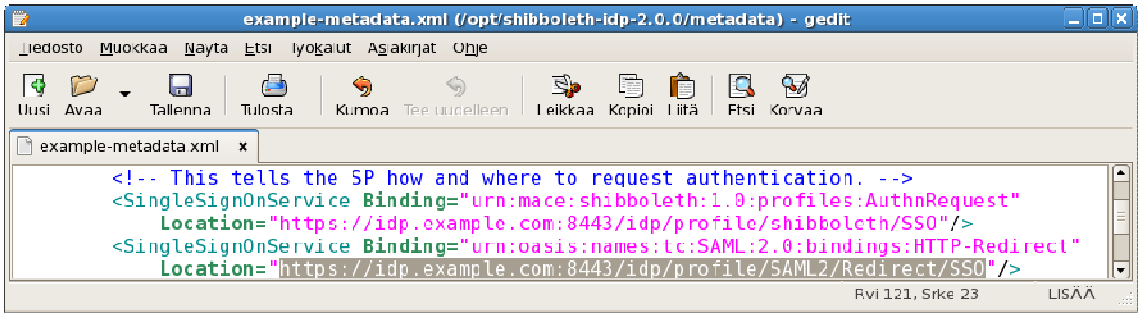
Kuvio 68: Metadatan muokkaus (3)



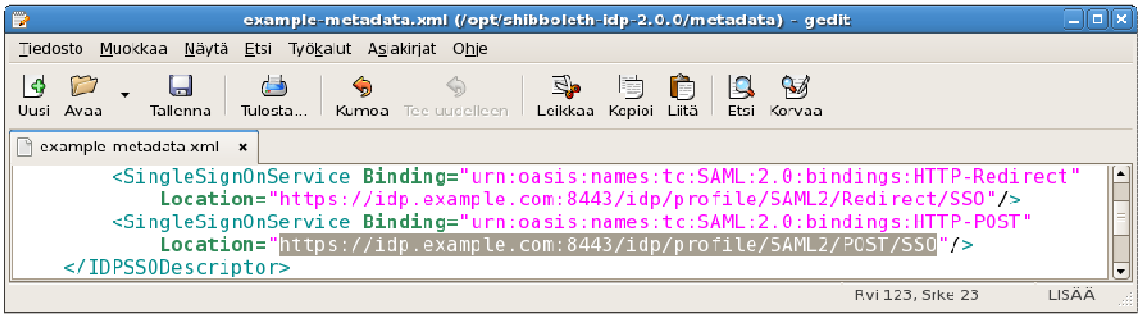
Kuvio 69: Metadatan muokkaus (4)

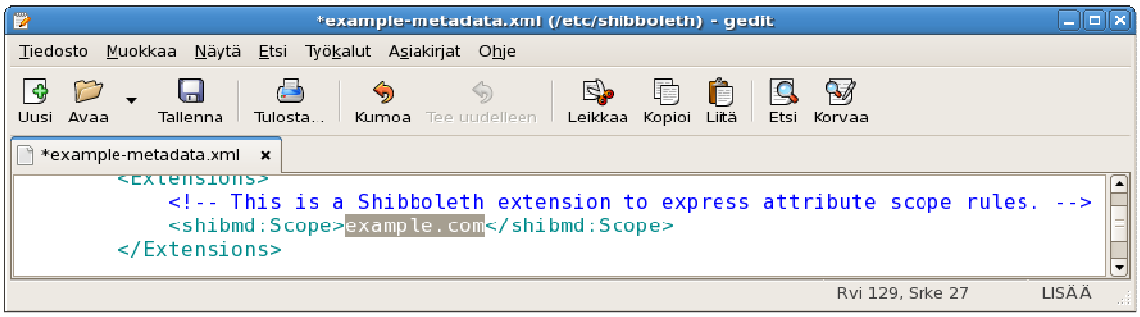


Kuvio 70: Metadatan muokkaus (5)

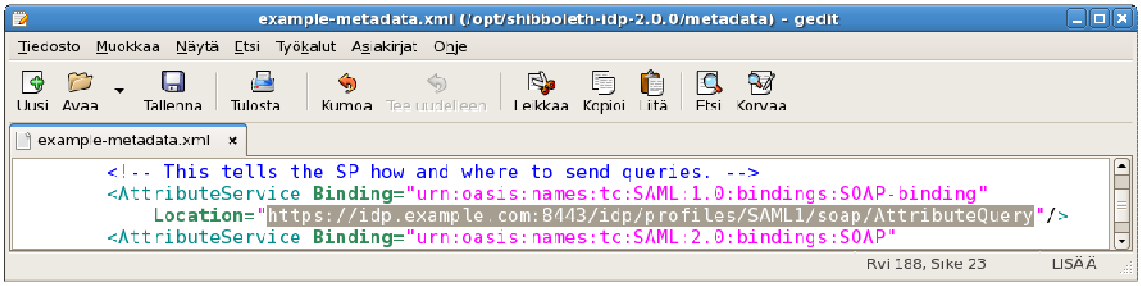


Kuvio 71: Metadatan muokkaus (6)

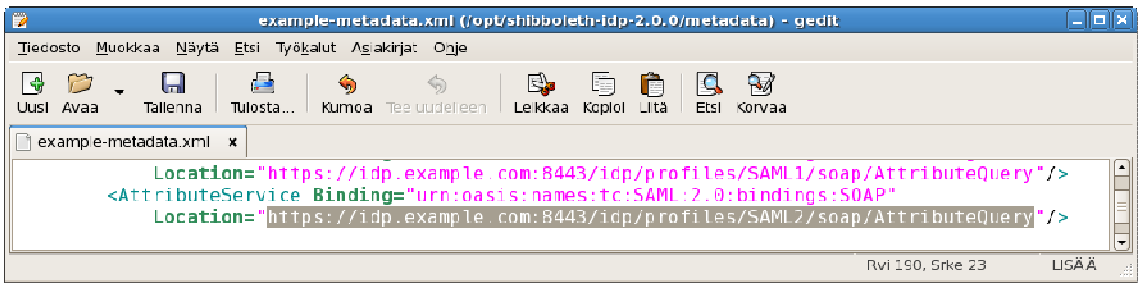




Kuvio 72: Metadatan muokkaus (7)



Kuvio 73: Metadatan muokkaus (8)



Kuvio 74: Metadatan muokkaus (9)

Näiden muutosten jälkeen halutessaan voi vielä muuttaa <Organization> ja <Contactperson> koodien sisällä olevat tiedot eli organisaatiota ja ylläpitäjää koskevat tiedot. Nämä eivät kuitenkaan ole järjestelmän toimivuuden kannalta oleellisia.

Muutosten jälkeen tallennetaan tiedosto. Kopioidaan valmis example-metadata.xml muistitikulle Service Providerin koneelta. Tämän jälkeen kopioidaan example-metadata.xml tiedosto muistitikulta Identity Providerin koneelle hakemistoon /opt/shibboleth-idp-2.0.0/metadata/

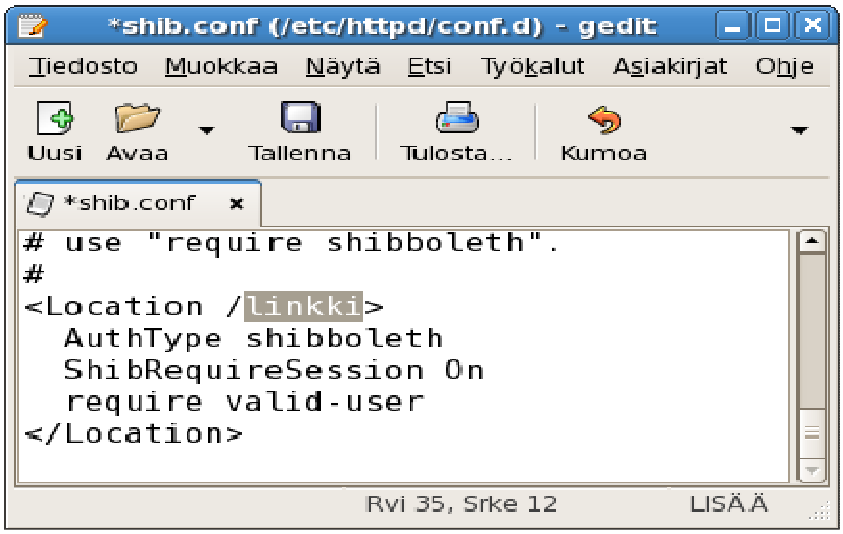
8.8 Service Provider-konfigurointi

Käymme jokaisen Service providerin konfiguroitavan tiedoston erikseen läpi.

8.8.1 Shibd.conf

Avataan tekstieditoriin shibd.conf tiedosto, joka löytyy hakemistosta /etc/httpd/conf.d/

Suojataan Apachen oletuskansiossa sijaitseva linkki hakemisto shibbolethilla tekemällä tiedostoon seuraava muutos:



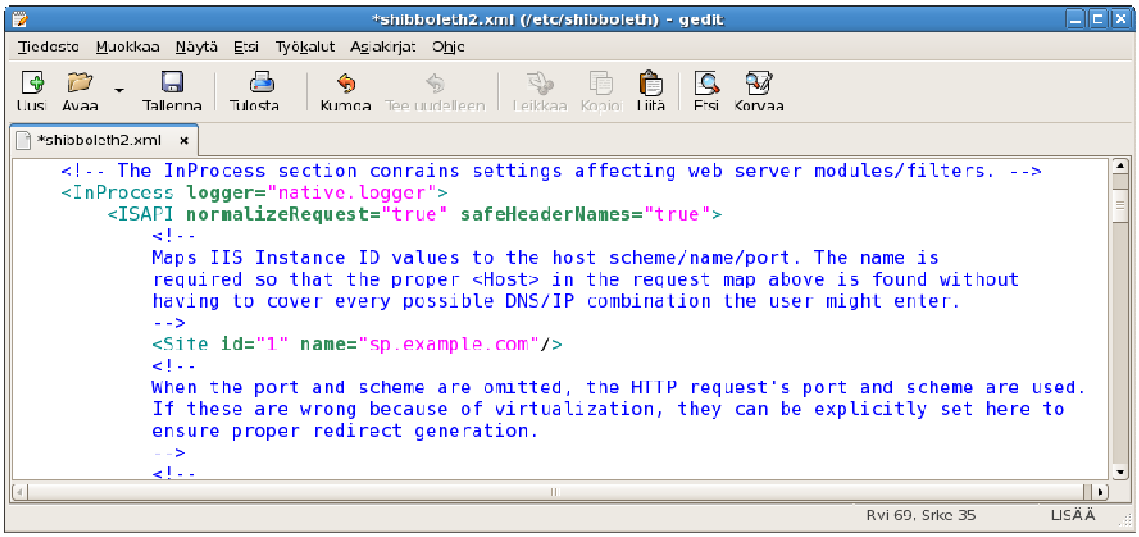
Kuvio 75: Linkin suojaus Shibbolethilla

8.8.2 Shibboleth2.xml

Shibboleth2.xml on Service Providerin pääkonfigurointitiedosto. Tiedoston sisällä on kommenteissa kuvattu mitä tiedoston jokainen osio tekee ja konfiguroi.

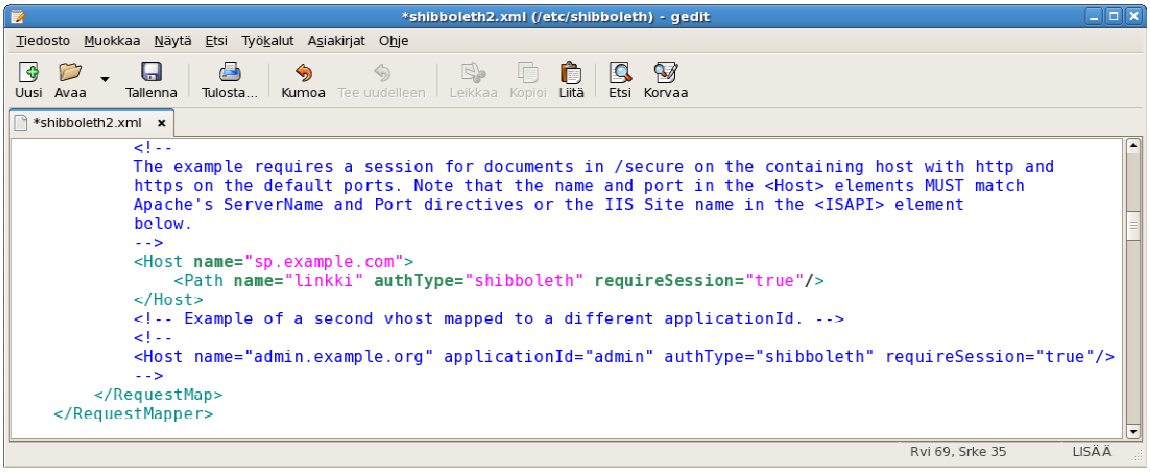
Avataan tekstieditoriin Shibboleth2.xml tiedosto joka löytyy hakemistosta /etc/shibboleth/

Muokataan Site name oikeaksi:



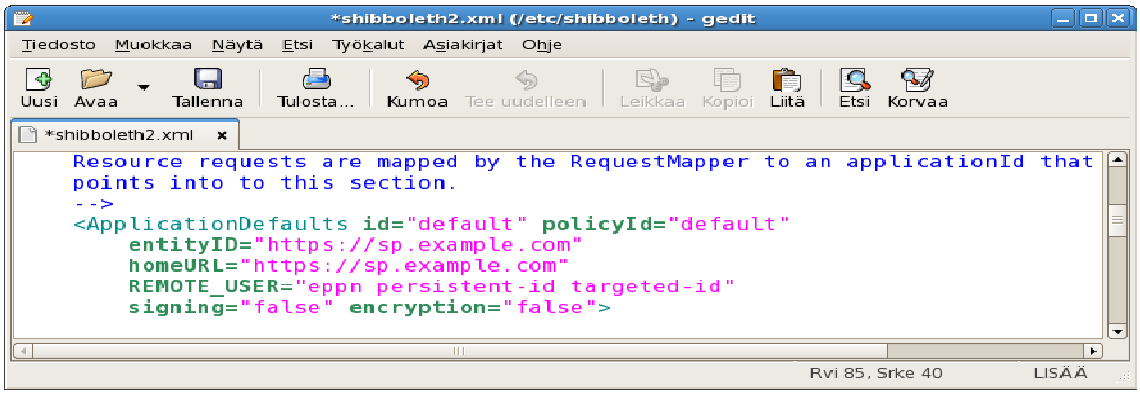
Kuvio 76: Shibboleth2.xml muokkaus (1)

Muokataan Host name ja Path name oikeiksi:



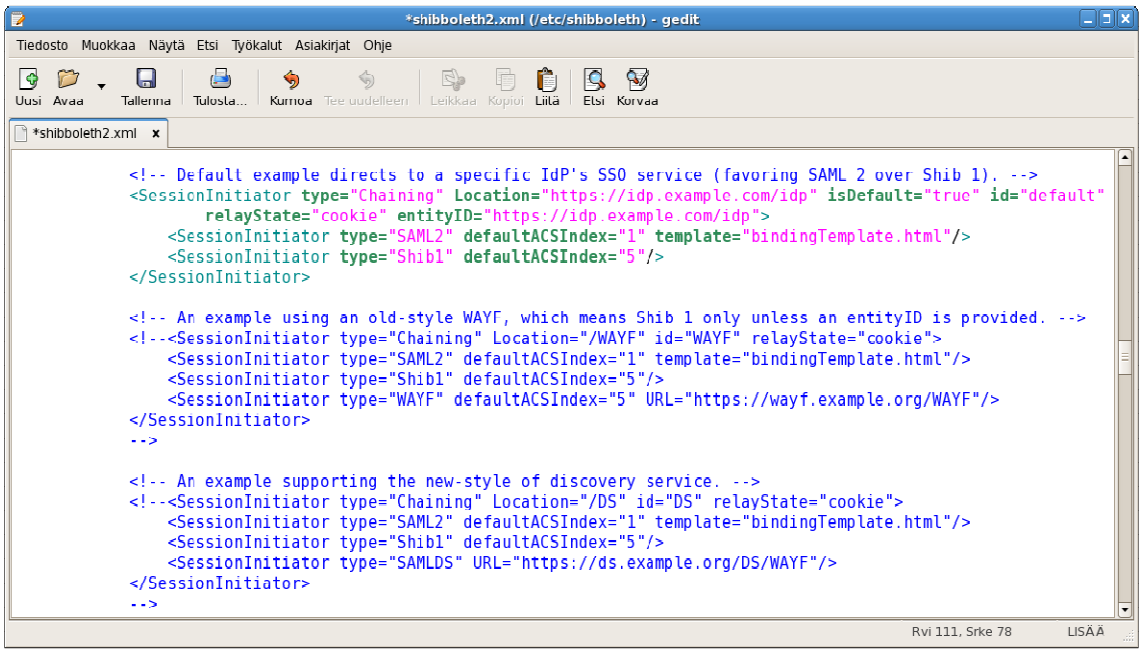
Kuvio 77: Shibboleth2.xml muokkaus (2)

Muokataan EntityID ja homeURL oikeiksi:



Kuvio 78: Shibboleth2.xml muokkaus (3)

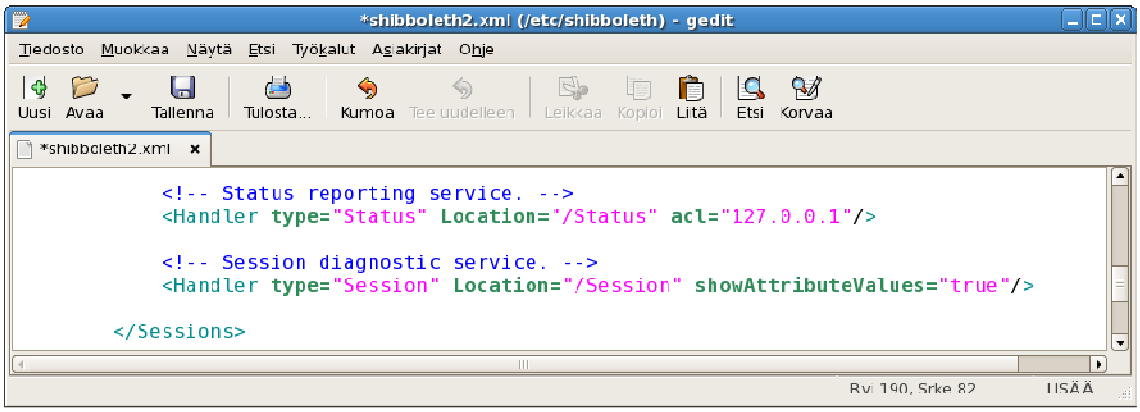
Kommentoidaan <!—ja → merkeillä kaksi viimeistä sessioninitiatoria(WAYF,DS) pois joita ei käytetä ja muokataan ensimmäinen sessioninitiator ottamaan yhteys IdP:hen muokkaamalla se seuraavanlaiseksi:



Kuvio 79: Shibboleth2.xml muokkaus (4)

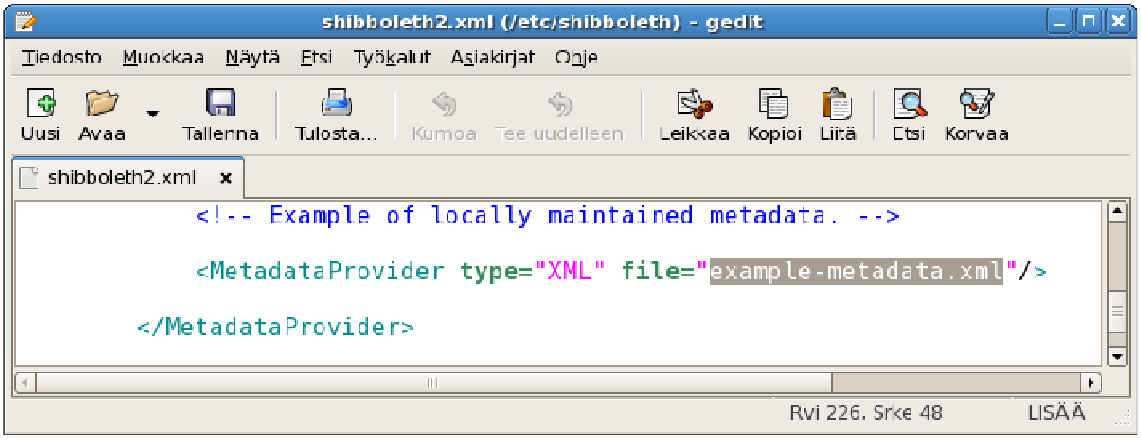
WAYF tai DS sessioninitiatoria voi halutessaan käyttää poistamalla kommentit ja muokkaamalla url osoitteet haluamiinsa. WAYF ja/tai DS mahdollistavat Service Providerin ottaa yhteys discovery serviceen, josta voi valita missä Identity Providerissa haluaa tunnistautua.

Muokataan Service Provider näyttämään Identity Providerilta saadut attribuutit session tiedoissa antamalla showAttributeValuesille arvo true :



Kuvio 80: Shibboleth2.xml muokkaus (5)

Otetaan kommentit pois locally maintained metadata Metadataprovider kohdalta ja muokataan se käyttämään tehtyä metadataa:



Kuvio 81: Shibboleth2.xml muokkaus (6)

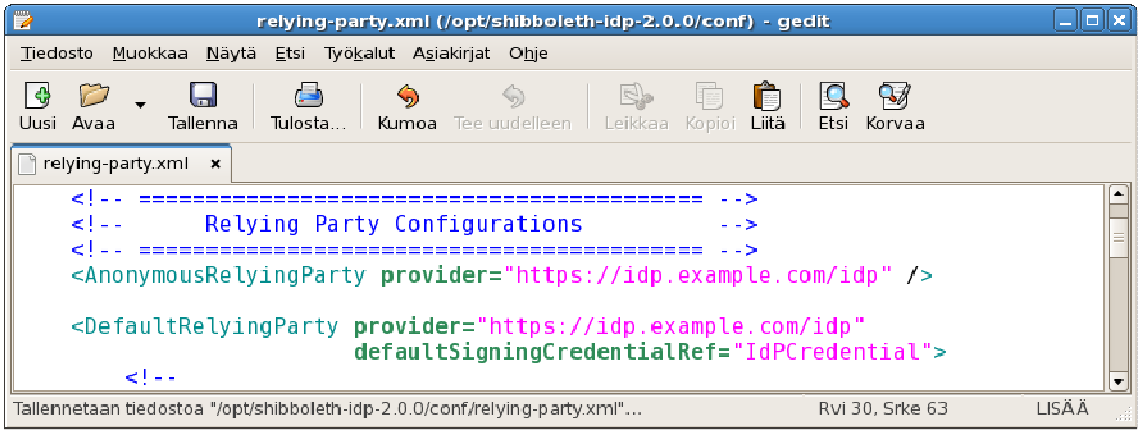
8.9 Identity Providerin konfigurionti

Käymme jokaisen Identity Providerin konfiguroitavan tiedoston erikseen läpi.

8.9.1 Relying-party.xml

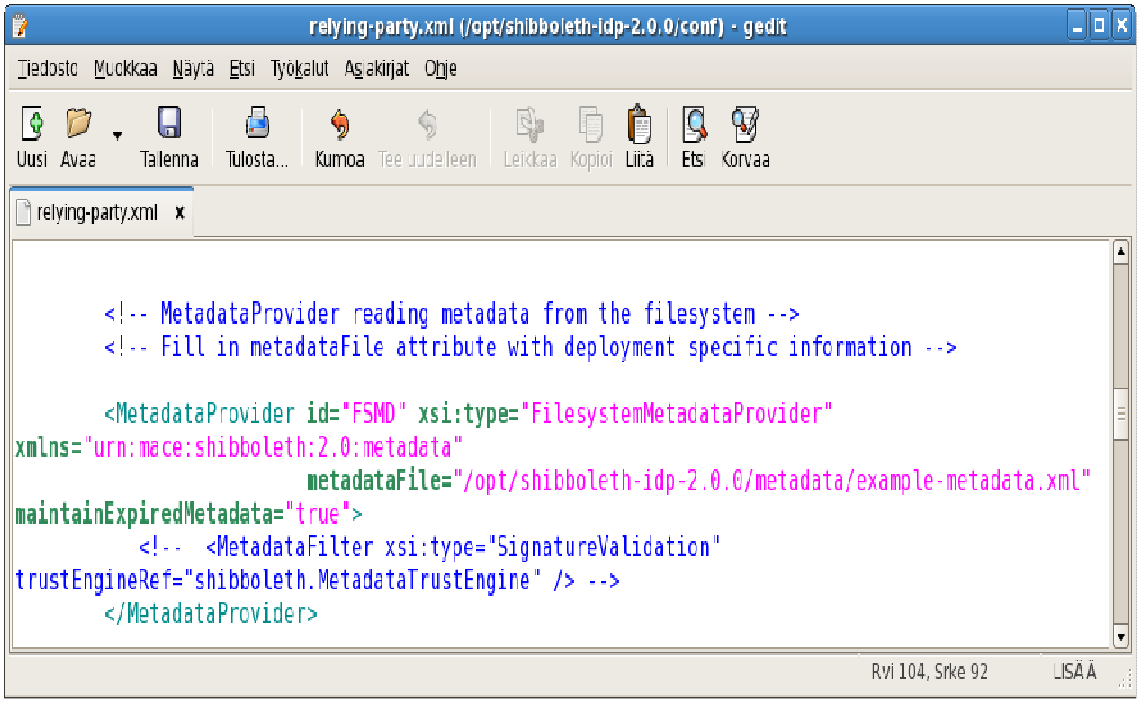
Relying-party.xml tiedostossa määritetään miten kyseinen Identity Provider käyttäytyy eri Service Providereita kohtaan. Tässä asennuksessa emme tee erityissääntöjä omalla Service Providerillemme, vaan asetamme Identity Providerin hakemaan vain perustiedot metadatasta.
Relying-party.xml löytyy hakemistosta /opt/shibboleth-idp-2.0.0/conf/
Avataan tiedosto tekstieditoriin.

Annetaan oikeat nimet AnonymousRelyingParty ja DefaultRelyingParty kohtiin:



Kuvio 82: Relying-party konfigurointi (1)

Laitetaan Identity Provider käyttämään tehtyä metadataa. Otetaan kommentit pois filesystem MetadataProviderin ympäriltä ja annetaan metadataFile kohtaan oikea sijainti. Lisäksi kommentoidaan MetadataFilter pois käytöstä:

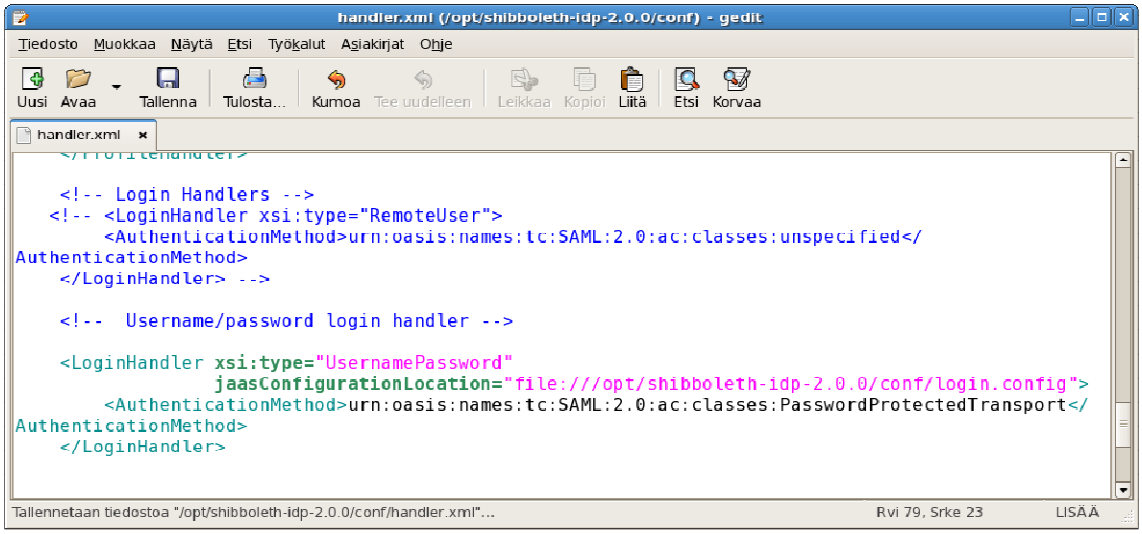


Kuvio 83: Relying-party konfigurointi (2)

8.9.2 Handler.xml

Handler.xml tiedostossa määritämme login handlerit, eli mitä metodia Shibboleth käyttää käyttäjän tunnistamiseen. Tässä asennuksessa käytämme autentikointia LDAP-hakemistoa vasten. Handler.xml löytyy hakemistosta /opt/shibboleth-idp-2.0.0/conf/
Avataan tiedosto tekstieditoriin.

Otetaan RemoteUser LoginHandler pois käytöstä laittamalla kommentit sen ympärille. Otetaan käyttöön UsernamePassword LoginHandler ottamalla kommentit pois sen ympäriltä:

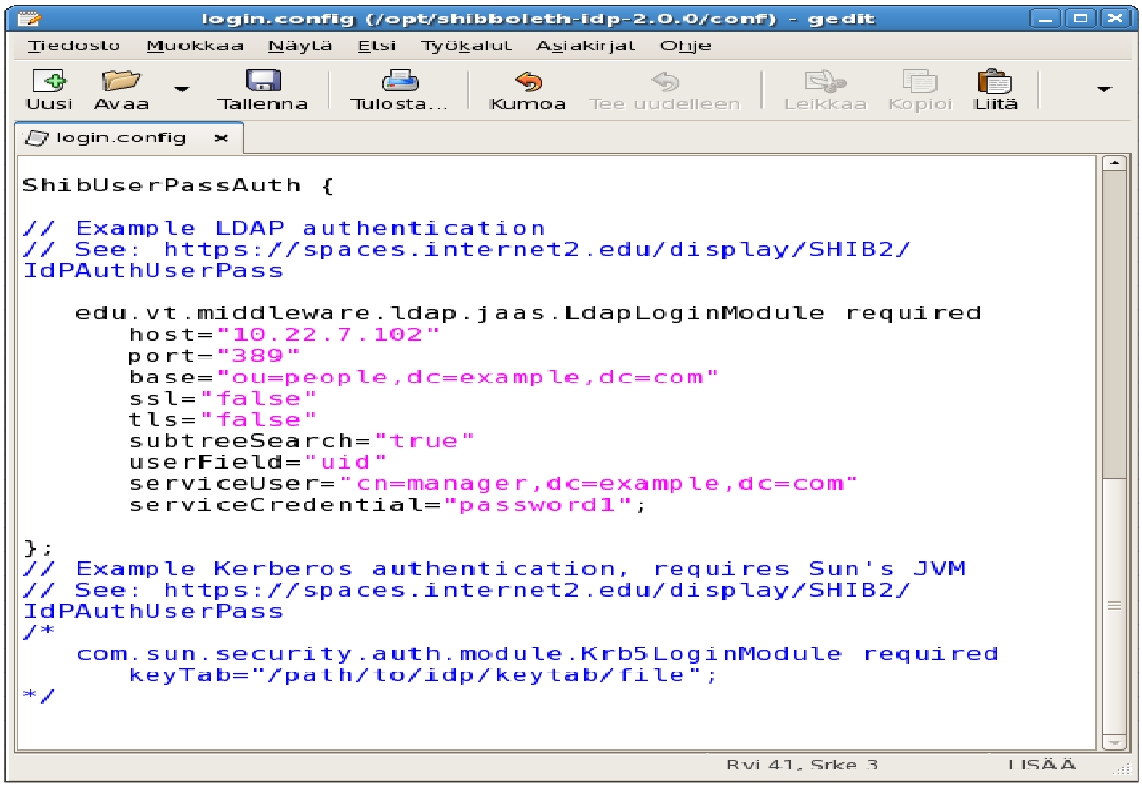


Kuvio 84.handler.xml konfigurointi

8.9.3 login.config

login.config tiedostosto on JAAS autentikointimoduulin konfigurointitiedosto. Tässä määritämme mihin LDAP:iin JAAS ottaa yhteyden. Määritetään JAAS ottamaan yhteys luomaamme LDAP-hakemistoon. login.config tiedosto löytyy hakemistosta
/opt/shibboleth-idp-2.0.0/conf/

Otetaan kommentit pois ennen edu.vt.middleware.ldap.jaas.LdapLoginModulea ja userfieldin jälkeen. Siirretään }; merkki paikalleen userfieldin jälkeen. Muokataan ja lisätään seuraavat muutokset:



Kuvio 85: Login-config muokkaus

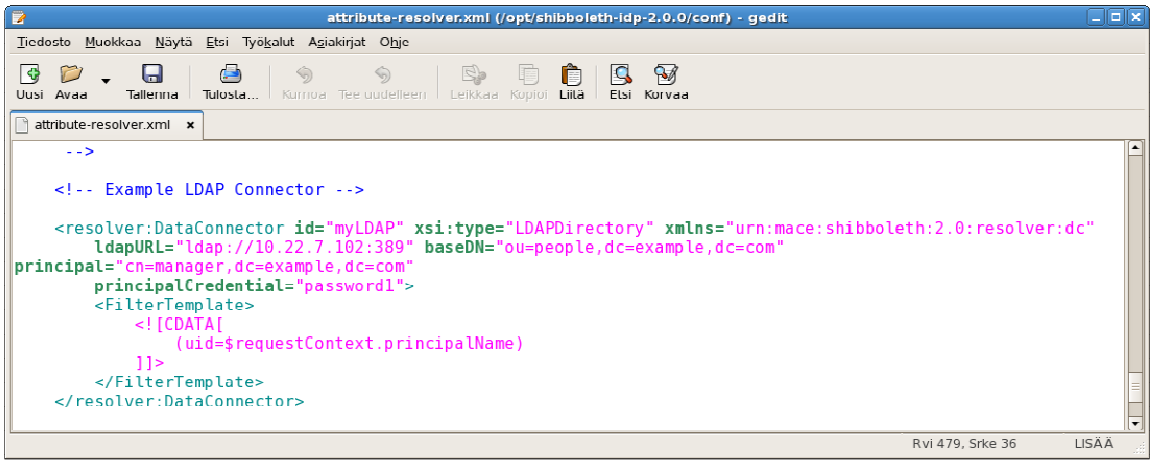
8.10 Identity Providerin attribuuttien konfigurointi

Käymme jokaisen Identity Providerin attribuutteihin liittyvän konfigurointitiedoston erikseen läpi.

8.10.1 attribute-resolver.xml

Attribute-resolver.xml tiedostossa määritämme mitä attribuutteja Identity Provider hakee ja mistä. Tiedosto sijaitsee hakemistossa /opt/shibboleth-idp-2.0.0/conf/

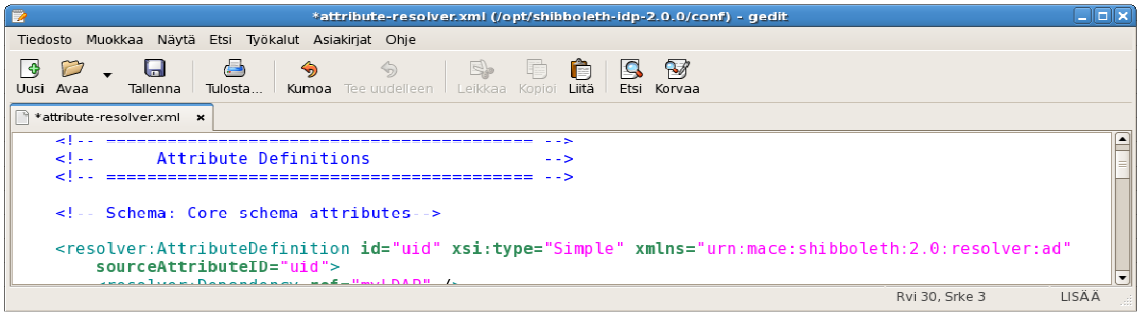
Määritetään Identity Provider hakemaan attribuutit luomastamme LDAP-hakemistosta. Muokataan LDAP DataConnector järjestelmämme mukaiseksi. Otetaan kommentit pois LDAP data-connectorin ympäriltä. Muokataan ldapURL, baseDN, principal ja PrincipalCredential oikeiksi:



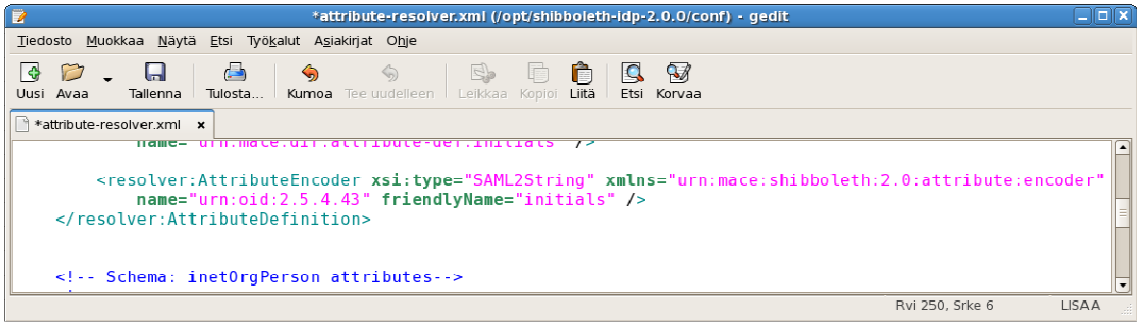
Kuvio 86: Attribute-resolver.xml

DataConnectorissa id määrittää, millä nimellä kyseistä LDAP-connectoria voi käyttää Shibbolethissa, ldapURL määrittää osoitteen, BaseDN määrittää, mistä ldapin hakemistosta attribuutteja etsitään, principal määrittää käyttäjän, jolla on oikeus käyttää ldappia ja principaCredential määritetyn käyttäjän salasanan.

Otetaan Core schema attribuutit käyttöön jotta Shibboleth osaa hakea niitä. Otetaan kommentit pois Core schema attribuuttijonon ympäriltä:



Kuvio 87: Kommentien poisto Core Scheman ympäriltä



Kuvio 88: Kommentien poisto Core Scheman ympäriltä

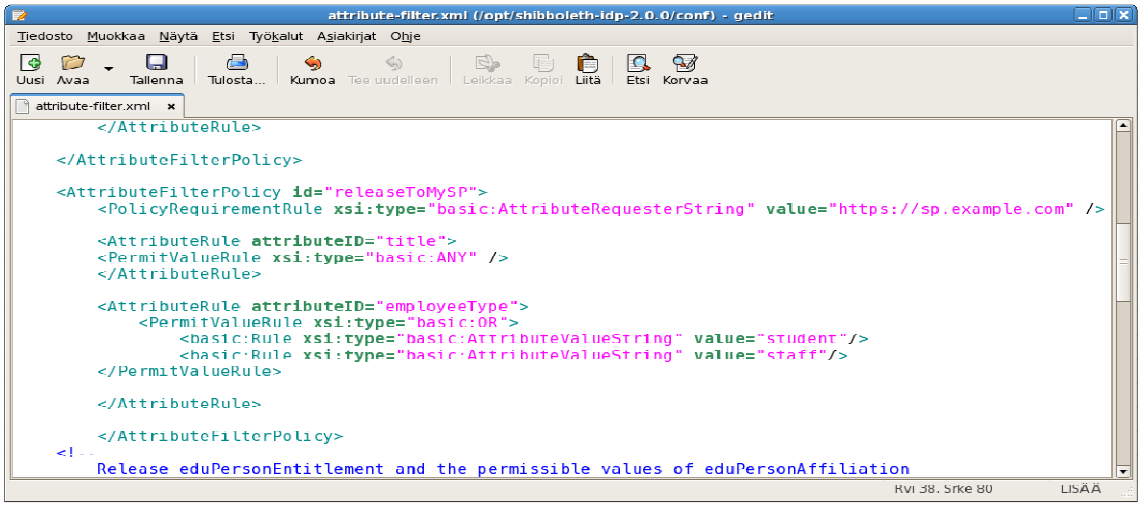
Tehdään sama operaatio inetOrgPerson attribuuteille.

Attribute-resolver.xml tiedostosta löytyy valmiina Core schema, InetOrgPerson sekä eduPerson attribuutteja. Käytämme tässä asennuksessa Core schema sekä inetOrgPerson attribuutteja, halutessaan muut attribuutit voi ottaa käyttöön poistamalla kommentit niiden ympäriltä. AttributeDefinition kohdassa id määrittää millä nimellä Identity Provider käyttää haettua attribuuttia ja sourceAttributeID määrittää millä nimellä attribuuttia haetaan tietokannasta. Dependency määrittää mistä tietokannasta attribuuttia haetaan, tässä tapauksessa myLDAP jonka olemme antaneet ldap hakemistomme id:ksi. AttributeEncoder kohdissa name määrittää millä nimellä Identity Provider lähettää attribuutin eteenpäin.

8.10.2 attribute-filter.xml

Attribute-filter.xml tiedostossa määritämme mitä attribuutteja Identity Provider saa lähettää eteenpäin ja kenelle. Voimme siis konfiguroida IdP:n lähettämään tietyt attribuutit tietylle Service Providerille ja estää sitä lähettämästä niitä jollekkin muulle SP:lle. Voimme myös määrittää millä attribuutin arvolla sen saa lähettää eteenpäin ja millä ei.

Määritämme Identity Providerin lähettämään title ja employeeType attribuutit Service Providerillemme. Määritämme myös employeeType attribuutille arvot jotka saa lähettää eteenpäin. Luodaan uusi AttributeFilterPolicy, joka määrittää edellämainitut asiat, luomme sen tiedostossa olemassa olevan AttributeFilterPolicyn jälkeen:



Kuvio 89: Attribuuttien muokkaus

AttributeFilterPolicyn id määrittää policyn nimen, type määrittää mitä sääntöä käytetään ja value säännön arvon. Tässä tapauksessa käytämme sääntöä joka määrittää mille Service Providerille Policyn attribuutit saa lähettää ja annamme sille arvoksi Service Providerimme metadatassa määritetyn EntityID:n. Määritämme ensimmäisellä AttributeRulella että attribuutin title jokaisen arvon saa lähettää ja toisella Attributerulella määritämme että attribuutin employeeType arvot student ja staff saa lähettää eteenpäin. Jokaselle attribuutille joka halutaan lähettää eteenpäin, pitää tehdä AttributeRule ja sisällyttää se AttributeFilterPolicyn sisään. Lisää sääntöjä löytyy Shibboleth Wikin internet sivuilta.

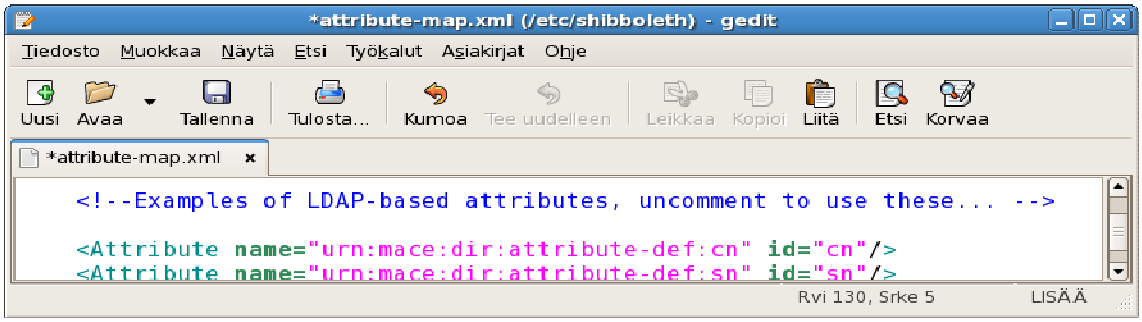
8.11 Service Providerin attribuuttien konfigurointi

Käymme läpi erikseen jokaisen Service Providerin attribuutteihin liittyvän konfigurointitiedoston.

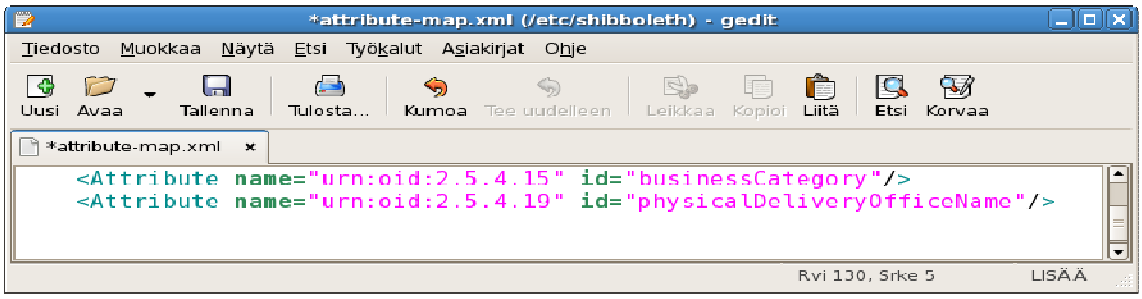
8.11.1 Attribute-map.xml

Attribute-map.xml tiedostossa määritämme mitä attribuutteja Service Provider ottaa vastaan ja millä nimellä. Attribute-map.xml tiedosto löytyy hakemistosta /etc/shibboleth/

Otetaan LDAP-based attribuutit käyttöön poistamalla kommentit niiden ympäriltä:

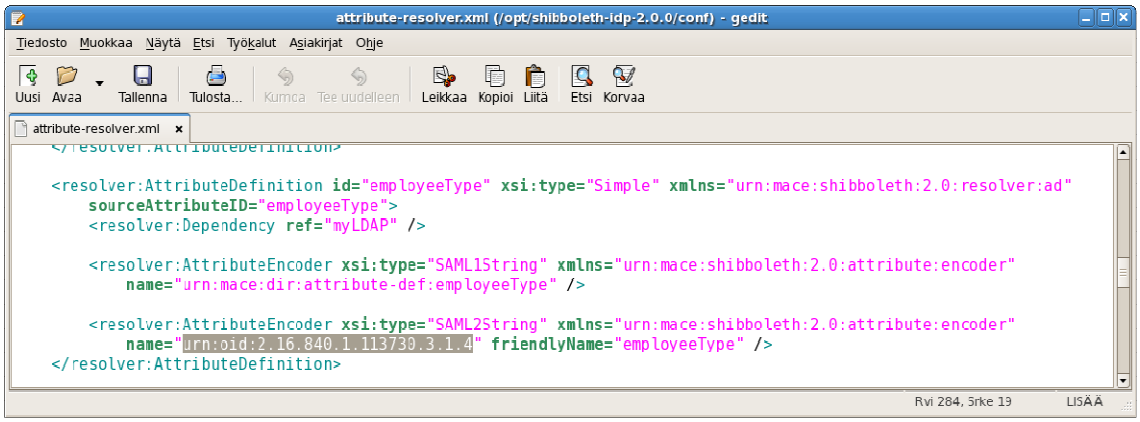


Kuvio 90: Attribuuttien käyttöönotto (1)



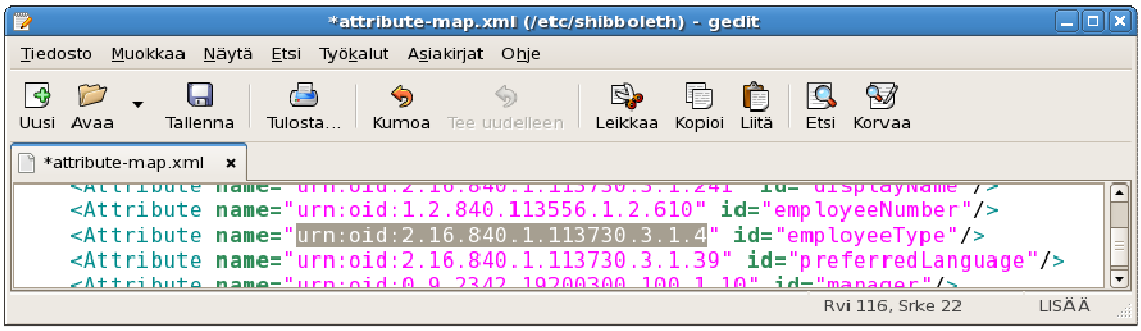
Kuvio 91: Attribuuttien käyttöönotto (2)

EmployeeType attribuutin name on asennuksessamme Service Providerilla eri kuin Identity Providerilla, joten muutamme sille oikean nimen. Oikea nimi saadaan Identity Providerin attribute-resolver.xml tiedostosta. Otamme SAML2 nimen koska olemme määrittäneet Shibbolethin lähettämään attribuutit SAML2 muodossa. Katsotaan oikea nimi:



Kuvio 92: SAML nimikkeen tarkistus

Muutetaan oikea nimi attribute-map.xml tiedostoon:



Kuvio 93: SAML nimekkeen muuttaminen

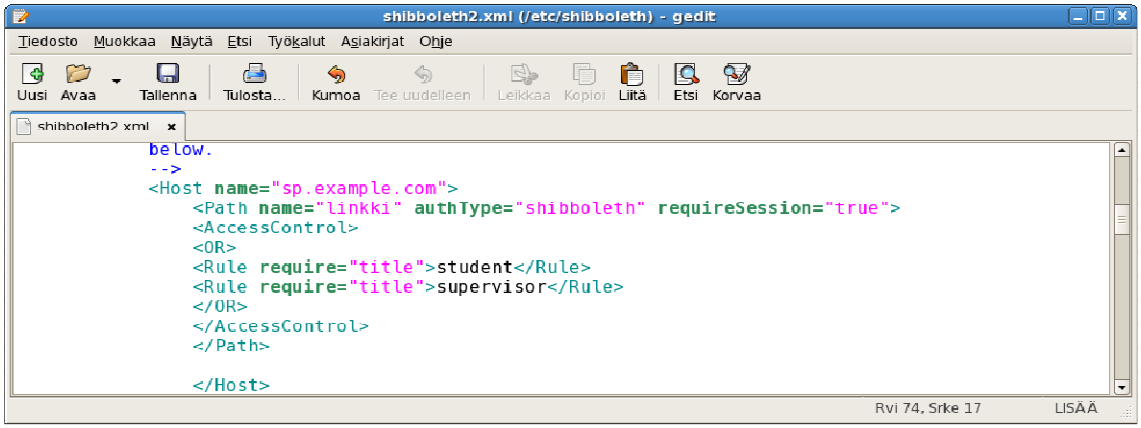
Attribuutteja lähetettäessä ja vastaanotettaessa täytyy aina tarkistaa että Identity Providerin ja Service Provider attribuuteilla antamat nimet täsmäävät toisiinsa.

8.11.2 Attribute-policy.xml

Attribute-policy.xml tiedostossa voimme määrittää jokaiselle attribuutille erikseen säännöt, miten niitä käytetään. Attribuutista voidaan esim. tarkistaa onko attribuutin scope eli attribuutti@example.com attribuutin @ merkin jälkeinen osa sama kuin metadatassa on määritetty kyseiselle Idp:lle josta attribuutti on saatu. Tiedostossa tulee valmiina käytössä oleva sääntö joka ei määritetä erityissääntöjä millekkään attribuutille. Emme tee erityissääntöjä attribuuteille tässä asennuksessa.

8.11.3 Shibboleth2.xml

Shibboleth2.xml tiedostossa voimme määrittää attribuutteja kulunvalvonnan käyttöön. Voimme määrittää millä tietyn attribuutin arvoilla pääsee sisään suojattuun palveluun. Asetamme Shibbolethin päästämään palveluun sisään henkilöt joiden title attribuutti on joko student tai supervisor. Muokkaamme RequestMapperin Host kohdan seuraavanlaiseksi:

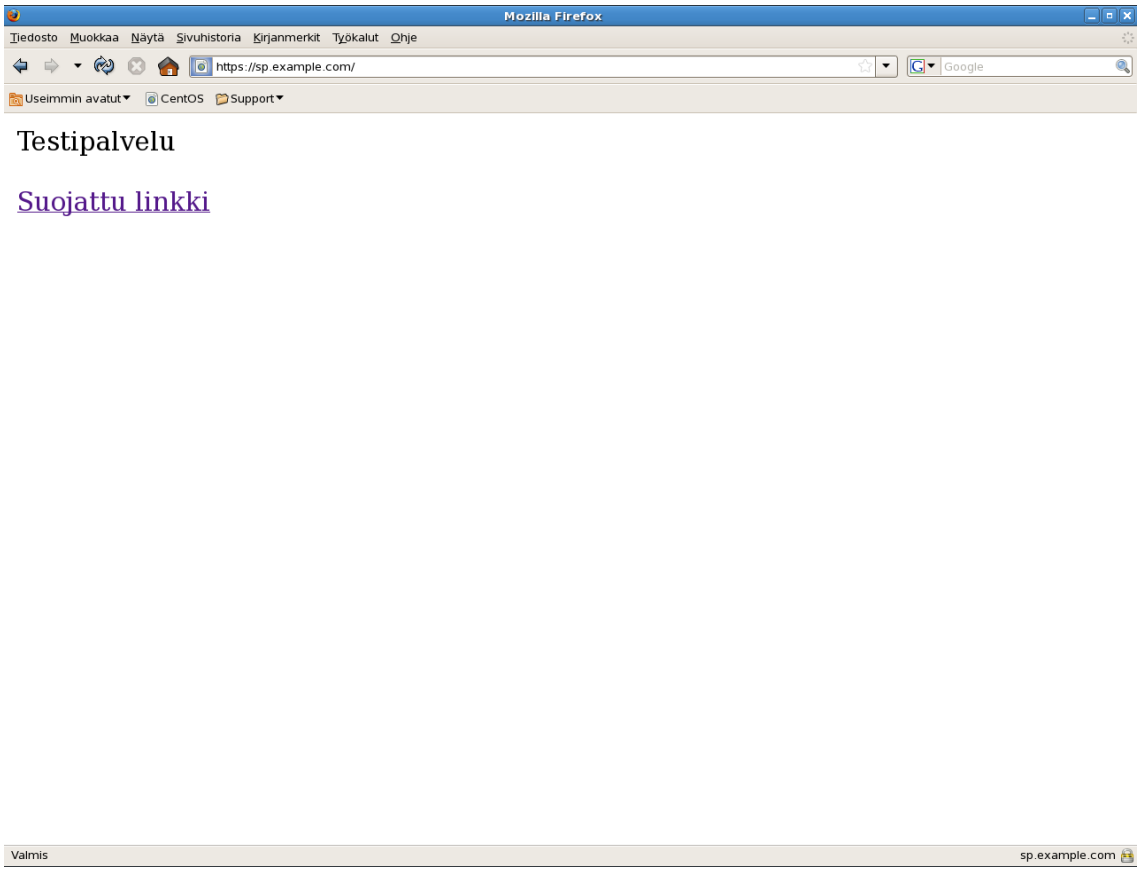


Kuvio 94: Shibboleth2.xml

Edellisessä laitoimme AccessControl elementin sisälle säännön joka vaatii attribuutilta title arvoa student tai supervisor päästääkseen käyttäjän suojattuun palveluun.

8.12 Valmiin asennuksen toiminta

Siirrytään Identity Providerin tai Service Providerin koneella osoitteeseen
https://sp.example.com

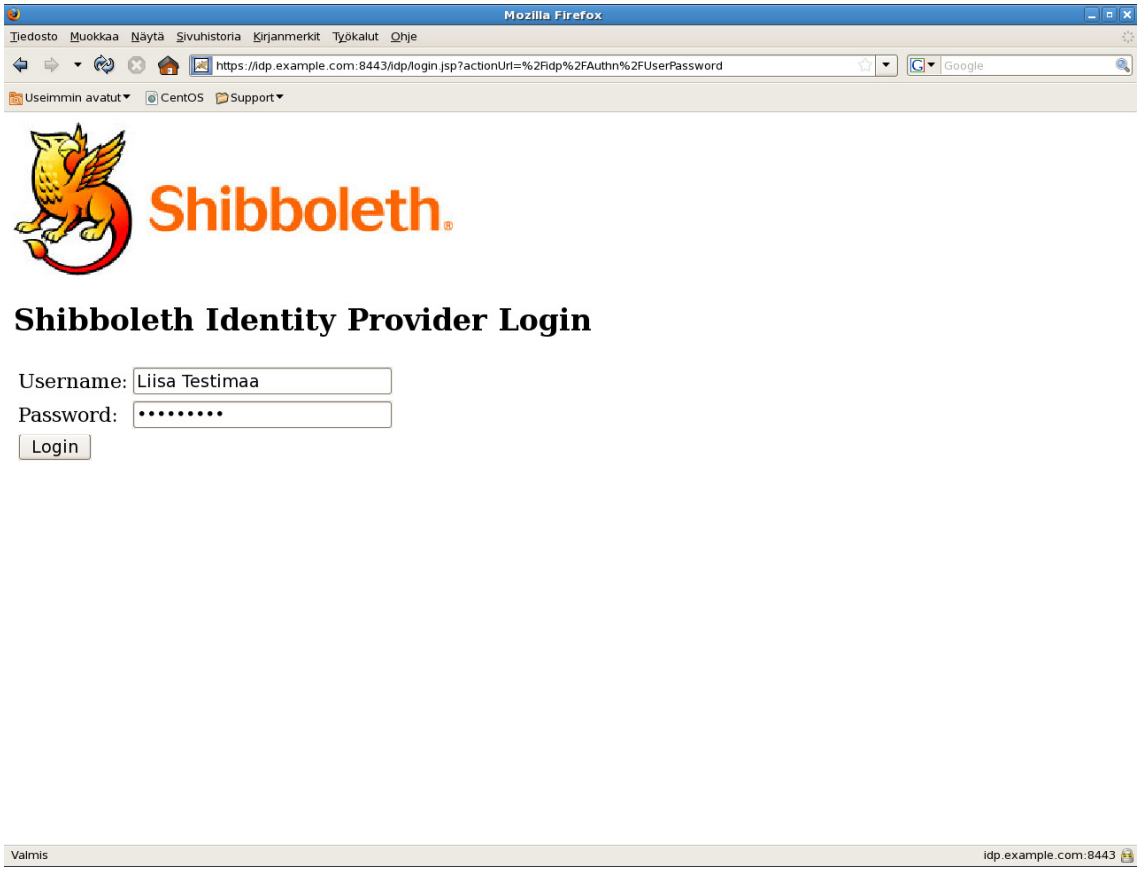


Kuvio 95: Aloitussivu

Osoitteen voi avata muillakin samassa verkossa olevalta koneelta, edellyttäen että koneen host-tiedostoon on määritetty Shibbolethin järjestelmän IP-osoitteita vastaavat nimet.

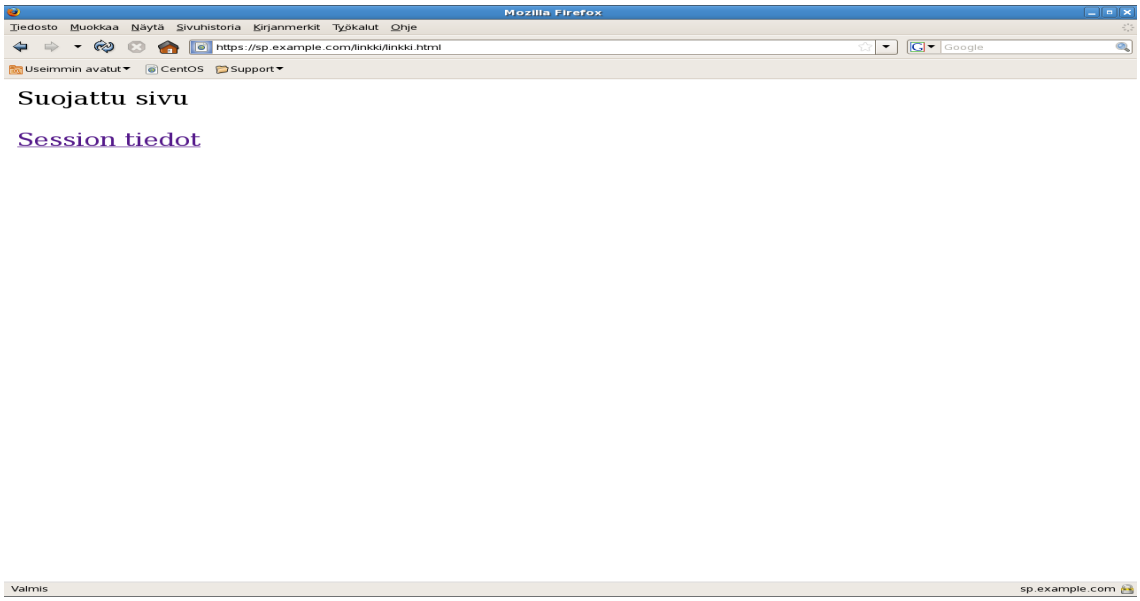
Painetaan linkkiä Suojattu linkki, jolloin Shibboleth käynnistyy tunnistaessaan että haettu sisältö on suojattua.

Kirjoitetaan Username kohtaan joku kolmesta aiemmin ldap hakemistoon määritetystä käyttäjästä ja annetaan sitä vastaava salasana:



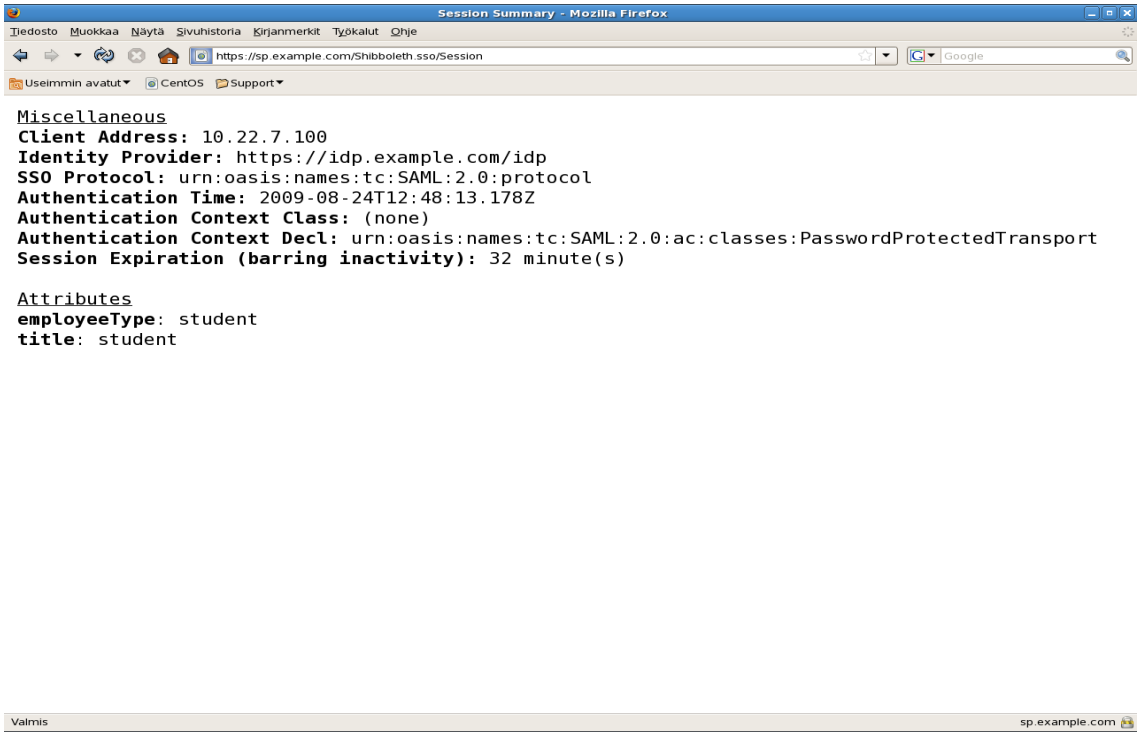
Kuvio 96: Tunnusten syöttö

Shibboleth autentikoi käyttäjän ldap hakemistosta, hakee käyttäjän attribuutit ja päästää käyttäjän suojattuun sisältöön:



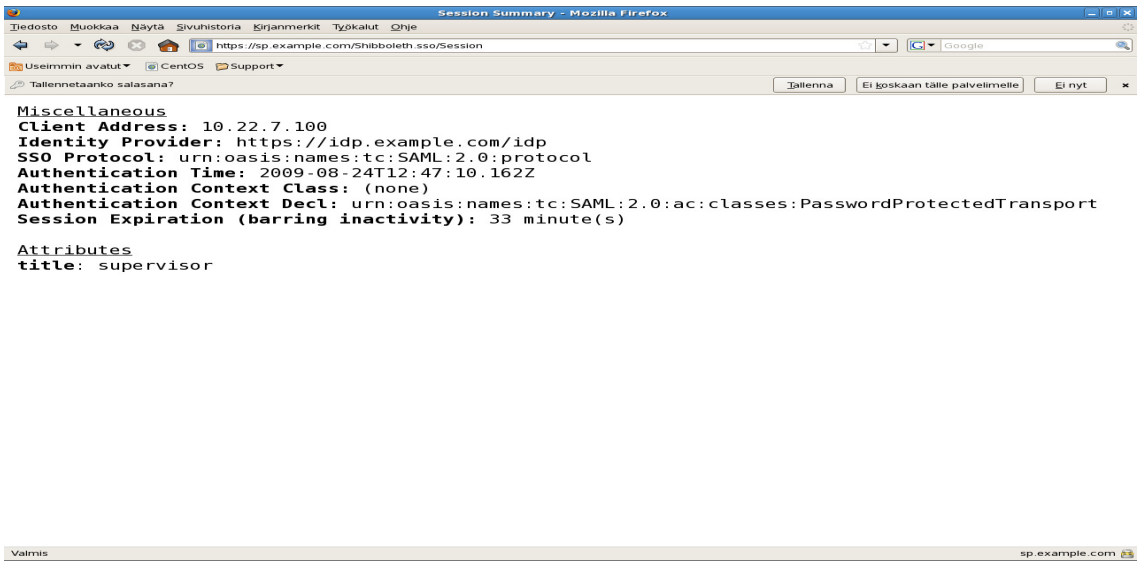
Kuvio 97: Suojattu sisältö

Painamalla linkkiä Session tiedot saamme näkyviin Shibboleth sessionin tiedot, alapuolella käyttäjän Liisa Testimaa tiedot:



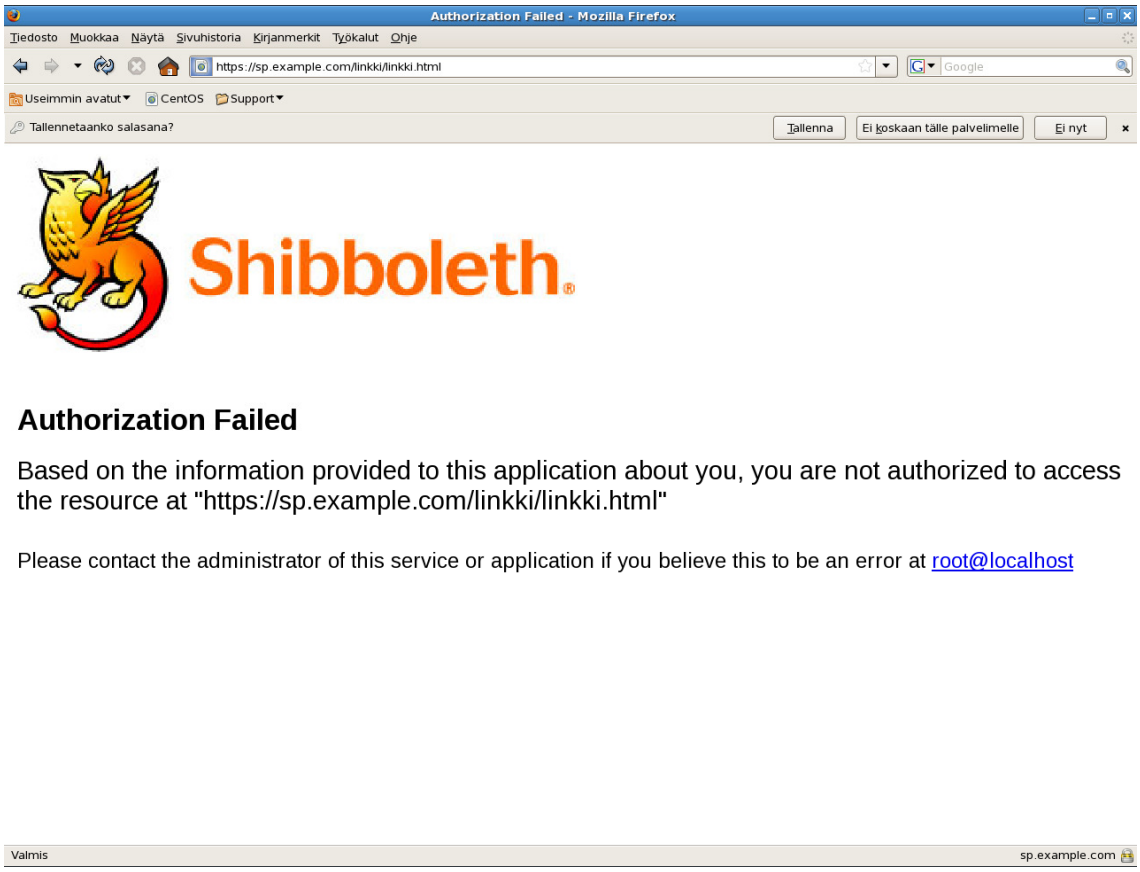
Kuvio 98: Shibboleth session tiedot

Session tiedot käyttäjästä Pentti Penaali:



Kuvio 99: Shibboleth session tiedot (2)

Käyttäjä Matti Meikalainen ei pääse tekemiemme asetusten mukaan sisään järjestelmään, joten ruutuun tulee seuraava virheilmoitus yritettäessä kyseisellä käyttäjällä palveluun:



Kuvio 100: Epäonnistunut kirjautuminen

9 Johtopäätökset

Shibboleth-järjestelmä on yleisesti ottaen toteutettu Linux-alustalle. Tarjolla ei ole vastaavia pilotointeja Windowsille, joten puuttuvista ohjeista johtuen toteutus tapahtui Linux-alustalle. Toteutus Windowsille olisi ollut varmasti mielenkiintoista sekä palkitsevaa yhtä aikaa, mutta sen toteuttaminen olisi tuonut esiin aivan liian paljon riskejä jotka olisivat voineet kaataa projektin. Linux-käyttöjärjestelmän käyttö on monille hieman vierasta. Aluksi onkin syytä perehtyä sen käyttöön, esim. erilaisiin komentoihin. Lähtökohtaisesti käytössä on enimmäkseen linuxin tekstipohjainen käyttöliittymä, ”terminaali”, eikä graafinen käyttöliittymä. Komentojen ja muiden käskyjen osaaminen helpottaa järjestelmän asennusta ja sen ymmärrystä.

Avoimen lähdekoodin toteutuksissa on niin hyviä kuin huonoja puoliakin. Avoimen lähdekoodin ominaisuuksiin kuuluu että se on kaikkien vapaasti hyödynnettävissä ja muokattavissa. Ohjelmia voidaan kehittää, mutta samalla ohjelmiin voidaan sijoittaa tarkoituksellisesti tietoturva-aukkoja tai muita haitallisia koodinpätkiä jotka sitten taas rikkovat tietoturvallista eheyttä. Avoimen lähdekoodin toteutuksessa yleensä aivan virallisia ohjeita ole saatavilla, koska ohjelmat ikään kuin elävät jatkuvasti. Shibboleth-järjestelmään on olemassa muutamia eri organisaatioiden tekemiä asennusohjeita, joista mikään ei kuitenkaan kata koko järjestelmän selkeätä asennusta ja konfigurointia. Tärkein kulmakivi olikin tehdä asennus- ja konfigurointiohje, jota käyttämällä täysin asiaan perehtymätön saa järjestelmän toimimaan. Näin ison järjestelmän asennuksessa ongelmilta on mahdoton välttyä. Ongelmiin siis pitää osata varautua ja löytää ratkaisu niihin. Suurimpiin ongelmiin löytyykin varsin kattavasti neuvoja foorumeilta sekä eri organisaatioiden yhteisöistä. Osassa ongelmista on kuitenkin tukeuduttava Suomessa shibboleth-järjestelmään tukea antavaan tahoon, CSC:hen. Järjestelmä sisältää logitoiminnot, mutta kestää jonkin aikaa ennen kuin huomaa kuinka järjestelmän logit toimivat ja kuinka niitä voi hyödyntää. Virhetilanteissa logit näyttävätkin yleensä itse pulman, vaikka hieman epäselvään sävyyn. Osa ongelmista johtuu eri ohjelmistojen ja tietokirjastojen eri versioista; tietty ohjelmistoversio toimii tietyn tietokirjastoversion kanssa jne. Myös käyttöjärjestelmä Linux voi aiheuttaa ongelmia, sillä jotkut ohjelmistot eivät toimi tiettyjen Linux-distribuutioiden kanssa. Projektin Käytössä oli CentOS-distribuutio, sillä näyttäisi siltä ,että se tuo mukanaan vähiten yhteensopivuusongelmia.

Shibboleth-järjestelmä sopii hyvin korkeakoulujen ja muiden samantyyppisten instituutioiden yhdistäväksi federoiduksi tunnistautumisjärjestelmäksi. Kuitenkin jos järjestelmää haluttaisiin käyttää laajemmalla skaalalla, esim. koko Suomen kansallisena tunnistautumisjärjestelmänä, pitäisi sen tietoturvaa tutkia enemmän ja antaa sen hallinnointi kansallisesti luotettavan tahon käsiin.

10 Projektin vaiheet ja arvio

Shibboleth-järjestelmä kokonaisuudessaan oli täysin uusi projektitiimillemme, joten siihen perehtymiseen menikin oma aikansa. Tarkoitus oli kerätä mahdollisimman paljon tietoa Shibbolethiin liittyen, jotta työn varsinainen asennusosio kävisi mutkitta. Tieto ja taito kerättiin pääasiallisesti internet-lähteistä ja Shibboleth-wikistä.

Projektin päätavoite oli siis oppia Shibboleth-järjestelmän asennus sekä konfigurointi. Emme tyytyneet vain valmiiseen järjestelmään vaan pyrimme samalla myös ymmärtämään ja selittämään itsellemme, miksi juuri päädyimme tiettyyn ratkaisuun eri tilanteissa. Pyrimme myös tutustumaan hieman erilaisiin ei-kaupallisiin käyttäjätunnistustekniikoihin WWW:ssä sekä punnitsemaan mikä tekniikka tai menetelmä olisi siihen soveltuvuudeltaan paras. Shibboleth-projektin taustalla oli siis paljon muutakin kuin vain itse järjestelmän asennus.

Projektimme alkoi osallistumalla kahden päivän mittaiseen Shibboleth-koulutukseen Espoon Keilaniemessä. Koulutustilaisuuden järjesti CSC tietotekniikan koulutuskeskus. Osallistuimme tilaisuuteen Laurea-Ammattikorkeakoulun edustajina. Koulutuksessa käytiin läpi Shibboleth Installfest asennus sekä itse järjestelmän konfigurointi. Koulutuksen tavoitteena oli tutustuttaa käyttäjä kyseiseen järjestelmään ja opastaa sen käyttöönotossa. Kurssin tiukasta aikataulusta johtuen saimmekin vain pintaraapaisun järjestelmään

Shibboleth-järjestelmää varten meidän piti hankkia kaksi erillistä palvelinkonetta sekä varata näille omat osoitteensa. Kyseistä toimenpidettä ei ollut mahdollista toteuttaa kotiympäristössä. Projektiin hankittiin tarvittava laitteisto, jonka saimme lainaksi ilman kustannuksia Laurea-Ammattikorkeakoululta. Projektimme infrastruktuuri koostui kahdesta palvelinkoneesta sekä niille asetetusta yksityisverkosta. Laitehankintojen jälkeen käytettävä laitteisto piti saada vastaamaan toteutettavan projektin vaatimuksia.

Järjestelmän asennus ei sujunut ilman ongelmia. Heti alusta ongelmia tuotti Shibboleth-järjestelmään liitettävät komponentit, kuten Apache ja Tomcat. Näiden ohjelmistojen käyttöä ja hallintaa ei käyty läpi missään ohjemateriaaleissa joita tarkastelimme, joten ohjelmistoihin tutustuminen vei oman aikansa. Vaikka internetistä löytyy Shibboleth-järjestelmään asennusohjeita, mitkään niistä eivät ole kaiken kattavia. Asennuksen ja konfiguroinnin edetessä törmäsimme moneen ongelmaan joihin ei yksinkertaisesti löytynyt ratkaisua mistään ohjemateriaaleista. Selvitimme monta ongelmaa tarkastelemalla itse järjestelmää ja sen lokitiedostoja kunnes löysimme korjauksen ongelmiin. Joissakin ongelmissa otimme yhteyttä CSC:n helpdeskiin, josta saimmekin viittauksia ratkaisuihin. Pyrimme tekemään asennusohjeen niin että sen pohjalta asennuksen tekevä ylläpitäjä ei törmäisi ongelmiin asennuksen ja konfiguroinnin aikana.

Asennuksen ja konfiguroinnin aikana piti kiinnittää jatkuvasti huomiota siihen, mitä olimme tekemässä ja minkä takia. Näin opimme ymmärtämään järjestelmää jatkuvasti enemmän ja pystyimme itse selvittämään ongelmatilanteita. Konfigurointitiedostojen muokkaaminen ilman suurempaa ymmärrystä ei olisi antanut samaa tieto- ja taitotasoa jonka saavutimme.

11 KEHITYSEHDOTUKSIA

11.1 Shibboleth-järjestelmän graafinen käyttöliittymä

Shibboleth järjestelmä ja siihen liittyvät komponentit asennetaan pääsääntöisesti käsin komentoriviltä käsin. Graafista käyttöliittymää ei Shibboleth-järjestelmän konfigurointiin ole, mikä saattaa tuottaa ongelmia aloittelijoille. Kaikki järjestelmän asetukset tehdään erillisiin tiedostoihin, jotka sijaitsevat eri hakemistoissa. Tiedostot ovat tekstitiedostoja, jotka sisältävät xml-koodia. Shibboleth-järjestelmän käyttöönottavien tahojen pitää siis omata suhteellisen kattavat perustiedot näiltä osa-alueilta, jotta asennus ja konfigurointi kävisi mutkitta. Tähän kohtaan olisikin hyvä puuttua ja kehittää jokin toimiva ratkaisu. Meidän mielestämme olisi vähintäänkin hyvä idea toteuttaa graafinen käyttöliittymä, jolla olisi mahdollisuus tehdä muutoksia järjestelmään. Tämä helpottaisi ja nopeuttaisi todella paljon järjestelmän muuttamista ja ennen kaikkea sen oppimista. Graafisen käyttöliittymän ei välttämättä tarvitsisi olla kovinkaan monimutkainen ja hienolla maulla toteutettu, vaan kaikessa yksinkertaisuudessaan karsittu mutta toimiva ratkaisu. Käyttöliittymän toteutus ei tulisi olemaan suuri ongelma koska periaatteen mukaan muutoksia tehtäisiin edellä mainittuihin tekstitiedostoihin eli oletetussa käyttöliittymässä olisi vain viittaus itse tiedostoon jonne muutos sitten tulee tapahtumaan.

11.2 Shibboleth-järjestelmän soveltaminen eri ympäristöihin

Laurea-Ammattikorkeakoulussa on käytössä oma organisaation sisäinen langaton verkko. Verkko on yleisesti kaikkien käytössä eikä sitä ole rajoitettu, kuka tahansa voi käyttää verkkoa. Kehityskelpoinen ideamme onkin että Laurea-Ammattikorkeakoulun wlan-autentikaatio tapahtuisi Shibbolethin avulla opiskelijanumeron sekä salasanan mukaan. Näin samalla saataisiin verkko avoimeksi myös muissa organisaatioissa opiskeleville. Idean toteutus olisi sinällään yksinkertainen, mutta langattoman verkon autentikointimetodia pitäisi muuttaa. Esimerkkinä olisi, että kun käyttäjä yhdistää langattomaan verkkoon, avautuisi web-selaimeen kirjautumisruutu jonne käyttäjätunnuksen ja salasanan syötettyään olisi mahdollista kirjautua verkkoon. Tähän samaiseen kirjautumissivuun integroitaisiin Shibboleth-järjestelmä, jolloin käytössä olisi organisaation ylittävä kirjautuminen.

Lähteet

KIRJALLISUUSLÄHTEET

Howes T, Smith M, Good G. 2003. LDAP directory Services. toinen painos. Pearson Education Inc.

Hunt, C. 2002. TCP/IP network administration. kolmas painos. O'Reilly media.

Laaksonen M, Nevasalo T, Tomula K. 2006. Yrityksen tietoturvakäsikirja. Edita.

Ray, E. 2003. Learning XML. toinen painos. O'Reilly media.

Todorov, D. 2007. Mechanics of User identification and authentication. Taylor & Francis Group, LLC.

VERKKOJULKAISUT

The Apache Software Foundation 2009. Apache. Luettu 04.05.2009.
http://tomcat.apache.org/

The Apache Software Foundation 2009. Tomcat. Luettu 04.05.2009.
http://tomcat.apache.org/

Christopher Betts 2009. JXplorer. Luettu 20.07.2009.
http://jxplorer.org/

CSC Tietotekniikan keskus 2009. Luettu 27.09.2009
http://www.csc.fi

CSC. Käyttäjähallinto korkeakoulussa. Luettu 12.09.2009.
www.csc.fi/hallinto/haka/hankkeita/org_sis_kayttajahallinto/kato_kartoitus_tiivistelma.pdf

Internet2 2009. Shibboleth 2 documentation. Luettu 20.09.2009.
https://spaces.internet2.edu/display/SHIB/

Linden, M 2004. Käyttäjätunnistuksen tekniikoita www:ssä. Luettu 10.05.2009.
http://www.helsinki.fi/atk/lehdet/204/art14.htm

Novell 2007. Keskitetty käyttäjätunnistus yksinkertaistaa opiskelijahallintaa. Luettu 12.09.2009.
http://www.novell.com/linux/solutions/securityandidentity/securecampaign/docs/Case_LTY_Atea.pdf

OpenLDAP Foundation 2008.OpenLDAP. Luettu 04.05.2009.
http://openldap.org

Cibernarium 2009. SSL. Luettu 15.09.2009
http://www.cibernarium.tamk.fi/tietoturva2/SSL_yhteys.htm

SWITCH 2009. Luettu 20.09.2009
http://www.switch.ch/

SWITCH 2009. Luettu 20.09.2009
http://www.switch.ch/aai/demo/2/expert.html

The Community Enterprise Operating System 2009. CentOS. Luettu 04.05.2009.
http://www.centos.org/

University of Washington 2007. PubCookie. Luettu 28.09.2009
http://www.pubcookie.org/

Oasis 2004. SAML. Luettu 12.05.2009.
http://www.oasis-open.org/committees/download.php/6193/sstc-saml-tech-overview-1.1-draft-04.pdf

JULKAISEMATTOMAT LÄHTEET

Jylhä, P. 2004. Web-pohjainen käyttäjätunnistus. Opinnäytetyö
luettu 20.08.2009. http://octopus.oamk.fi/ropeliweppi/?sivu=download&id=753

Santala, N. 2008. Käyttäjätunnistuksen ongelmia ja illuusio tietoturvasta. Pro Gradu
luettu 10.08.2009. http://www.cs.uta.fi/research/theses/masters/Santala_Niko.pdf

Kuvat